DESIGN GUIDE

1.9.0 | June 2015 | 3725-33186-001A

Polycom® SoundStructure® C16, C12, C8, and SR12

Copyright© 2015, Polycom, Inc. All rights reserved. No part of this document may be reproduced, translated into another language or format, or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Polycom, Inc.

6001 America Center Drive
San Jose, CA 95002
USA

Trademarks Polycom®, the Polycom logo and the names and marks associated with Polycom products are trademarks and/or service marks of Polycom, Inc. and are registered and/or common law marks in the United States and various other countries.

All other trademarks are property of their respective owners. No portion hereof may be reproduced or transmitted in any form or by any means, for any purpose other than the recipient's personal use, without the express written permission of Polycom.

Disclaimer While Polycom uses reasonable efforts to include accurate and up-to-date information in this document, Polycom makes no warranties or representations as to its accuracy. Polycom assumes no liability or responsibility for any typographical or other errors or omissions in the content of this document.

Limitation of Liability Polycom and/or its respective suppliers make no representations about the suitability of the information contained in this document for any purpose. Information is provided "as is" without warranty of any kind and is subject to change without notice. The entire risk arising out of its use remains with the recipient. In no event shall Polycom and/or its respective suppliers be liable for any direct, consequential, incidental, special, punitive or other damages whatsoever (including without limitation, damages for loss of business profits, business interruption, or loss of business information), even if Polycom has been advised of the possibility of such damages.

End User License Agreement By installing, copying, or otherwise using this product, you acknowledge that you have read, understand and agree to be bound by the terms and conditions of the End User License Agreement for this product. The EULA for this product is available on the Polycom Support page for the product.

Patent Information The accompanying product may be protected by one or more U.S. and foreign patents and/or pending patent applications held by Polycom, Inc.

Open Source Software Used in this Product This product may contain open source software. You may receive the open source software from Polycom up to three (3) years after the distribution date of the applicable product or software at a charge not greater than the cost to Polycom of shipping or distributing the software to you. To receive software information, as well as the open source software code used in this product, contact Polycom by email at OpenSourceVideo@polycom.com.

Customer Feedback We are striving to improve our documentation quality and we appreciate your feedback. Email your opinions and comments to DocumentationFeedback@polycom.com.

Polycom Support Visit the Polycom Support Center for End User License Agreements, software downloads, product documents, product licenses, troubleshooting tips, service requests, and more.

Contents

Introduction 14

Introducing the Polycom SoundStructure Product Family 17

Defining SoundStructure Architectural Features 17
Understanding Polycom OBAM™ - One Big Audio Matrix 19
Understanding SoundStructure C-Series Products 21
Understanding C-Series Input Processing 22
Creating C-Series Matrix Crosspoints 28
Understanding C-Series Output processing 29
Processing C-Series Submixes 30
Understanding C-Series Acoustic Echo Canceller References 31
Understanding SoundStructure SR-Series Products 32
Understanding SR-Series Input Processing 34
Creating SR-Series Matrix Crosspoints 41
Understanding SR-Series Output Processing 42
Processing SR-Series Submix 43
Understanding Telephony Processing 44

Introducing SoundStructure Design Concepts 47

Understanding Device Inputs and Outputs 47
Understanding Physical Channels 48
Numbering Physical Channel On A Single SoundStructure Device 48
Numbering Physical Channel With Multiple SoundStructure Devices 49
Physical Channel Summary 55
Understanding Virtual Channels 56
Virtual Channel Summary 57
Understanding Virtual Channel Groups 58
Virtual Channel Group Summary 62
Understanding Telephone Virtual Channels 62
Defining Logic Pins 62
Labeling Physical Logic Pins 63
Controlling Virtual Channels 67

    Controlling Array Virtual Channels 68
  Understanding IR Receiver Virtual Channel 69

Creating Designs with SoundStructure Studio 70
  Understanding SoundStructure Studio 70
    Understanding System Requirements 71
    Viewing Recommended Operating System 71
    Installing SoundStructure Studio 71
    Operating in Online and Offline Mode 77

Customizing SoundStructure Designs 80
  Using the Wiring Page 80
    Editing Devices 83
  Using the Channels Page 84
    Editing Virtual Channels 87
    Creating Virtual Channel Groups 88
  Setting Input Signals 91
    Enabling Input Signal Meters 92
    Using Input Channel Controls 97
    Operating Analog Signal Gain 98
    Changing the Mute Status 99
    Enabling Phantom Power 99
    Using the Ungated Type 100
    Using Delay Type 104
    Using Delay Compensation 106
    Using Trim 107
    Processing Equalization 108
    Eliminating Feedback 110
    Enabling Acoustic Echo Cancellation (AEC) 112
    Processing Noise Cancellation 113
    Using Automatic Gain Control (AGC) 115
    Using Dynamics Processors 116
    Using Automatic Microphone Mixing 121
    Processing Delay 125
    Controlling Fader 125
  Defining a Signal Generator 126
  Setting Output Signals 128
    Processing Output Dynamics 129
    Processing Output Equalization 129
    Processing Delay 131

Processing Submix Signals . . . 131
  Processing Output Dynamics . . . 132
  Processing Output Equalization . . . 132
  Processing Delay . . . 134
  Controlling Fader . . . 134
Using the Matrix Page . . . 134
  Adjusting Crosspoints . . . 136
  Matrix summary . . . 141
Using the Telephony Channels . . . 142
  Adjusting Input Gain . . . 143
  Processing Noise Cancellation . . . 144
  Using Automatic Gain Control (AGC) . . . 144
  Processing Output Dynamics . . . 145
  Processing Equalization . . . 145
  Controlling Fader . . . 146
  Processing Delay . . . 146
  Using Telephone Controls . . . 147

Connecting Over Conference Link2 . . . 150
Connecting SoundStructure Conference Link2 . . . 150
Integrating Polycom Video Codec . . . 151
Designing with The Polycom Video Codec . . . 153
  Editing The Polycom Video Codec Input Channels . . . 153
  Processing The Polycom Video Codec SoundStructure Signals . . . 156
  Understanding The Polycom Video Codec Output Channels . . . 157
  Routing The Polycom Video Codec Signals . . . 158
  Using the Mute Controls . . . 159
  Using the Volume Controls . . . 161
Designing With Polycom Digital Microphone Arrays . . . 163
  Understanding Digital Microphone Cabling Requirements . . . 164
  Updating Digital Microphone Firmware . . . 164
  Detecting CLink2 Devices . . . 166
  Viewing Digital Microphone Array Example . . . 166
  Assigning Digital Microphone Array Channels To Physical Inputs . . . 169
  Numbering Digital Microphone Array . . . 172
Understanding Installation Options . . . 174
Summary . . . 177

Linking Multiple SoundStructure Devices with One Big Audio Matrix . . . 179
Introduction . . . 179

Preparing Units for Linking with OBAM 179
    Updating SoundStructure Device Firmware 179
    Linking SoundStructure Devices 180
Creating a Multi-Device Configuration File 184
    Expanding or Contracting an Existing Project 185
    Creating a New Project 187
Uploading Configuration Files 190
    Controlling the SoundStructure System 193
    Accessing SoundStructure Logs 194
    Connecting Polycom Microphones 195
    Connecting Multiple Polycom Video Codec Conferencing Systems 198

Installing SoundStructure Devices 201

Configuration Files 201
Wiring The Devices 201
Uploading A Configuration File 205
Downloading A Configuration File 209
Updating Firmware 209
Configuring The Signal Gains 211
    Input Signal Level Adjustment 212
    Signal Meters 212
    Room Gain 214
    Telephony Signal Levels 216
    Output Signal Levels 218
Presets 223
    Preset Operation 224
    Saving Presets 225
    Creating Partial Presets 227
    Running Presets 231
    Removing Presets 232

Using Events, Logic, and IR 233

Understanding Events 233
    Sources 233
    Triggers 234
    Actions 234
Creating Events With SoundStructure Studio 236
    Adding New Events 236
    Enable And Disable Events 236
    Event Entries In The Logs 237

Removing Events With Studio 238
SoundStructure Studio Automatically Creates Events 238
Polycom IR Remote 242
Polycom IR Remote Channel ID 246
IR Receiver Connector 246
Logic Ports 247
Digital Logic Inputs 248
Analog Logic Inputs 250
Logic Outputs 250
Logic Arrays 251
Viewing Event Examples 253
Splitting and Combining Presets Triggered from a Logic Input 253
Viewing Push To Talk Microphones with LEDs Example 256
Viewing Push and Hold to Temporarily Mute A Microphone 260
Viewing the Phone Off Hook Drives A Relay Example 261
Viewing the Volume Knob Adjusts “Amplifier” Fader Example 263
Viewing the Gating Information Sent To A Control System Example 265
Positioning A Polycom Video Codec Camera Example 267
Creating SoundStructure Events Best Practices 268

Managing SoundStructure Systems 270
Connecting To The Device 270
LAN Interface 270
Dynamic IP Addresses 271
Link-Local IP Addresses 273
Static IP Addresses 274
Setting The Time Server 276
Control And Command Sessions 276
SoundStructure Device Discovery 278
AMX Beacon 278
RS-232 279
Configuring And Accessing The Logs 279

Integrating The Polycom Touch Control with SoundStructure Systems 281
Polycom Touch Control and SoundStructure Systems 281
SoundStructure System Requirements 281
Polycom Touch Control with Video CodecVideo Codec Systems vs. SoundStructure Systems 282
Pairing the Polycom Touch Control with SoundStructure 283
Polycom Touch Control Administrative Settings 287
Configuring the Polycom Touch Control LAN Properties 288

Configuring Polycom Touch Control Regional Settings 289
Configuring Security Options 290
Setting up Polycom Touch Control log management 290
Updating Polycom Touch Control Software 290
Using the Polycom Touch Control with SoundStructure 292
Designing a SoundStructure Project with the Polycom Touch Control 292
Using Multiple SoundStructure Telephony Interfaces 296
Using Multiple Polycom Touch Controls with SoundStructure 297
Validating Polycom Touch Control and SoundStructure integration 298

Integrating SoundStructure with SoundStructure VoIP Interface 302

Introduction 302
How to Read This Chapter 303
SoundStructure VoIP Interface Overview 303
SoundStructure System Requirements 305
Upgrading a Project to the SoundStructure VoIP Interface 305
Upgrading an Existing TEL1/TEL2 Project to the SoundStructure VoIP Interface 306
Creating a New Project with the SoundStructure VoIP interface 313
Upgrading the Firmware in the SoundStructure System 319
Installing the New Plugin Cards 321
Uploading the Configuration File 323
Configuring the SoundStructure VoIP Interface 324
Setting the IP address of the SoundStructure VoIP Interface 324
Setting the Provisioning Server settings 330
Registering Lines with the SoundStructure VoIP Interface 336
Using the SoundStructure VoIP Interface with SoundStructure Studio 341
Using the Phone Settings Control 341
Customizing SoundStructure Telephony Settings 344
SoundStructure VoIP Interface Settings on the Wiring Page 345
Setting an IP address with SoundStructure Studio 346
Using the SoundStructure Studio Console 349
Updating Software on the SoundStructure VoIP Interface 351
Upgrading Software with a Local FTP Server 351
Upgrading Software with an Existing Provisioning Server 352
Upgrading Software with the Web Configuration Utility 354
Validating a SoundStructure VoIP Interface Installation 361
VoIP Interface Logs 369
Back up and Restore the VoIP Specific Settings 371
Importing and Exporting VoIP Parameter Settings 373
SoundStructure Log Information 374

Information Required for Support 376
Understanding SoundStructure VoIP Interface API Commands 376
Using the SoundStructure API 379
  Dialing a Call 379
  Hanging up a Call 380
  Putting a Call on Hold and Resuming the Call 380
  Forwarding an Incoming Call 381
  Transferring a Call 381
  Blind Transfer of a Call 382
  Dialing Two Calls on the Same Line 383
  Dialing Two Calls on Different Lines 384
  SoundStructure API Behavior Changes 385

Adding Authentication to SoundStructure Systems 387

SoundStructure Authentication Overview 387
SoundStructure System Requirements 387
Enabling Authentication on a SoundStructure System 388
Discovering a System with Authentication 389
Removing Authentication from a SoundStructure System 391
Viewing SoundStructure Command Logs 392
Understanding SoundStructure System Compatibility Considerations 392
  SoundStructure Authentication API Command Summary 395

Creating Advanced Applications 397

Creating a One Microphone And Mono Video Conferencing System 397
  SoundStructure Studio Steps 398
  Using the Channels Page 401
  Using the Matrix Page 402
  Understanding Wiring Information 403
  Controlling The System 404
Creating Four Digital Array Microphones and A SoundStation VTX1000 Conferencing System . 405
  SoundStructure Studio Steps 407
  Editing Matrix Settings 410
  Editing Channels Settings 412
  Understanding Wiring Information 415
  Controlling The System 416
Creating an Eight Microphones, Video, and Telephony Application Conferencing System . 417
  Creating a Project in SoundStructure Studio 418
  Matrix Settings 422
  Channels Settings 424

Wiring Information 425
Controlling The System 426
Creating a Two PSTN Line Positional “Receive” Audio Conferencing System 427
SoundStructure Studio Steps 429
Matrix Settings 433
Channels Settings 436
Wiring Information 437
Controlling The System 438
Creating an Eight Microphones and Stereo Video Conferencing System 440
Channels Settings 445
Wiring Information 446
Controlling The System 447
Creating an Eight Microphones with The Polycom Video Codec Conferencing System 448
SoundStructure Studio Steps 448
Matrix Settings 452
Channels Settings 453
Wiring Information 454
Controlling The System 455
Creating an Eight Microphones with Wireless and Lectern Microphones Reinforcement Conferencing System 456
SoundStructure Studio Steps 458
Matrix Settings 459
Channels Settings 462
Wiring Information 470
Controlling The System 471
Creating a Sixteen Microphones with Six-Zone Sound Reinforcement Conferencing System . . . 472
SoundStructure Studio Steps 473
Matrix Settings 478
Channels Settings 481
Wiring Information 483
Controlling The System 485
Creating a Room Combining Application Conferencing System 485
SoundStructure Studio Steps 488
Combined Room Settings 492
Split Room Settings 496
Wiring Information 500
Controlling The System 501

Troubleshooting 503
Audio Troubleshooting 503

Echo Troubleshooting 505
API Troubleshooting 509
RS-232 Troubleshooting 512
Polycom Video Codec Integration 514
Telco Troubleshooting 515
Ethernet 515
Hardware Troubleshooting 516
OBAM Troubleshooting 517
Troubleshooting The IR Interface 518
Contacting Technical Support 518

Specifications 519
Technical Specifications 519
Pin Out Summary 521
  PSTN Cable 522
  Conference Link2 522
  OBAM Link 523
  IR Receiver 524
  RS-232 524
  Logic Interface 525
  Audio Connections 526

Using SoundStructure Studio Controls 528
Adjusting Knobs 528
Adjusting Matrix Crosspoints 528

Appendix A: Command Protocol Reference Guide 530
Using SoundStructure Command Protocols 530
Understanding SoundStructure Control Interfaces 530
  Understanding RS-232 530
  Connecting with the Ethernet Interface 531
  Using Virtual Channels 532
  Understanding Virtual Channel Types 533
  Understanding Virtual Channel Groups 534
Understanding SoundStructure Command Syntax 535
  Controlling SoundStructure Parameters 535
  Understanding the Command Format 537
  Understanding the Control Commands 538
  Understanding Virtual Channel Definition Commands 539
  Virtual Channel Group Definition Commands 543

- Adjusting Parameters 547
- Command List 551
  - Command Example 552
- SoundStructure Parameters 554
  - Gain and Mute Parameters 554
  - Matrix Parameters 561
  - Telephony Parameters 568
  - Equalizer Parameters 612
  - Dynamics Processing Parameters 631
  - Algorithm Parameters 649
  - Input Path Parameters 667
  - Automixer Parameters 669
  - GPIO Control Parameters 679
  - Control Port Parameters 682
  - System Parameters 692

Appendix B: Address Book 706
  - Using the Address Book 706
  - Address Book SoundStructure System Entries 707
  - Address Book Folders 710
  - Removing Entries from the Address Book 712
  - Changing the Location of the Address Book 712

Appendix C: Designing Audio Conferencing Systems 713
- Large Room Environments 714
- Microphone Selection And Placement 714
  - Microphone Fundamentals 714
  - Microphones For Conferencing 717
- Automatic Microphone Mixers 719
- Noise Cancellation 719
- Acoustic Echo Cancellation 720
  - AEC Reference 722
  - Tail Time 722
  - Transmission Delay 723
  - Echo Return Loss 724
  - Multi Channel vs. Single Channel AEC 725
  - Muting Microphones 725
  - Volume Control 726
  - AEC Troubleshooting Guidelines 726
- Telephone Hybrid 727

Amplifiers 728
Loudspeakers 728
    Speaker Zoning And Placement 730
    Loudspeakers - How Much Power Is Required 732
    Spatial Directionality 732
    Microphone And Loudspeaker Placement Considerations 732
In-Room Reinforcement 733

Introduction

The Polycom® SoundStructure® products are professional, rack-mountable, and audio processing devices that set a new standard for audio performance and conferencing in any style of room. With both monaural and stereo acoustic echo cancellation capabilities, the SoundStructure conferencing products provide an immersive conferencing experience that is unparalleled. The SoundStructure products are designed to integrate seamlessly with the Polycom Video Codec conferencing system and the Polycom® Touch Control for the ultimate experience with HD voice, video, content, and ease of use.

Note: Recent Product Name Changes Not Shown in Graphics

With the release of SoundStructure Firmware 1.7.0 and SoundStructure Studio 1.9.0, the product names for the Polycom video and microphone conferencing products have changed to reflect added support for Polycom® RealPresence® Group Series. However, the product name changes are not reflected in the graphics and screenshots shown in this guide. For example, although Polycom HDX is now Polycom Video Codec, some of the graphics in this guide still display the Polycom Video Codec as HDX.

Additionally, RealPresence Group Series is compatible with older versions of SoundStructure Studio and Firmware, and any concepts that refer to HDX apply for Group Series as well.

The Polycom SoundStructure C16, C12, and C8 audio conferencing devices are single rack unit devices that have 16 inputs and 16 outputs, 12 inputs and 12 outputs, or 8 inputs and 8 outputs respectively. The SoundStructure SR12 is an audio device for commercial sound applications that do not require acoustic echo cancellation capabilities and has 12 inputs and 12 outputs. Any combination of SoundStructure devices can be used together to build systems up to a total of eight SoundStructure devices and up to 128 inputs and 128 outputs. SoundStructure products can be used with any style of analog microphone or line-level input and output sources and are also compatible with the Polycom table and ceiling microphones.

The SoundStructure products are used in similar applications as Polycom’s Vortex® installed voice products but have additional capabilities including:

- Stereo acoustic echo cancellation on all inputs
- Direct digital integration with Polycom Video Codec or RealPresence® Group Series
- Feedback elimination on all inputs
- More equalization options available on all inputs, outputs, and submixes
- Dynamics processing on all inputs, outputs, and submixes
- Modular telephony options that can be used with any SoundStructure device
- Submix processing and as many submixes as inputs
- Ethernet port for configuration and device management

- Event engine for using internal state information such as muting, logic input and logic output ports, and an IR remote for controlling SoundStructure

SoundStructure devices are configured with Polycom's SoundStructure Studio software, a Windows®-based comprehensive design tool used to create audio configurations either online—connected to a SoundStructure system—or offline—not connected to a SoundStructure system. SoundStructure Studio is used to upload and retrieve configuration files to and from SoundStructure systems.

For detailed information on how to install a device, terminate cables, and connect other devices to the SoundStructure devices, refer to the SoundStructure Hardware Installation Guide. For information on the SoundStructure API command syntax used to configure SoundStructure devices and control the devices with third party controllers, refer to Appendix A: Command Protocol Reference Guide. The SoundStructure Command Protocol Reference Guide can also be found by pointing a browser to the SoundStructure device's IP address.

This guide is designed for the technical user and A/V designer who needs to use SoundStructure products, create audio designs, customize audio designs, and verify the performance of SoundStructure designs. This guide is organized as follows:

- Introducing the Polycom SoundStructure Product Family is an introduction to the SoundStructure products including the OBAM™ architecture and details of the signal processing available for inputs, outputs, telephony, and submix processing.
- Introducing SoundStructure Design Concepts presents the SoundStructure design concepts of physical channels, virtual channels, and virtual channel groups. These concepts are integral to making SoundStructure products easy to use and enable control system application code to be reused and portable across multiple installations.
- Creating Designs with SoundStructure Studio describes how to use the SoundStructure Studio windows software to create a design. Start with this section if you want to get up and running quickly using SoundStructure Studio.
- Customizing SoundStructure Designs provides detailed information on customizing the design created with SoundStructure Studio including all the controls presented as part of the user interface. Start with this chapter if you have a design and would like to customize it for your application.
- Connecting Over Conference Link2 provides information on the Conference Link2 interface and how SoundStructure devices integrate with the Polycom Video Codec conferencing system.
- Linking Multiple SoundStructure Devices with One Big Audio Matrix provides information on how to link multiple SoundStructure devices with the OBAM™ interface.
- Installing SoundStructure Devices provides information on how to install, set signal levels, and validate the performance of the SoundStructure devices. Start here if you have a system already up and running and would like to adjust the system in real-time.
- Using Events, Logic, and IR provides information on how to use SoundStructure 'events' with logic input and output pins, an IR remote, and for options for how to send commands from SoundStructure's RS-232 interface to other devices including a Polycom Video Codec.
- Managing SoundStructure Systems provides information for the network administrator including how to set IP addresses and how to view the internal SoundStructure logs, and more.
- Integrating The Polycom Touch Control with SoundStructure Systems provides the steps for using the Polycom Touch Control with a SoundStructure system. See the Users Guide for SoundStructure and the Polycom Touch Control for instructions on how to use the Polycom Touch Control with SoundStructure.
- Integrating SoundStructure with SoundStructure VoIP Interface provides the steps for designing with, and configuring, the SoundStructure VoIP interface.

- Adding Authentication to SoundStructure Systems introduces authentication and how to enable password protection on SoundStructure systems.
- Creating Advanced Applications provides example applications with SoundStructure products including stereo audio conferencing applications, room combining, and more.
- Troubleshooting provides troubleshooting information and steps including details on the status LEDs on SoundStructure.
- Specifications lists the Specifications for the SoundStructure devices including audio performance, power requirements, and more.
- Using SoundStructure Studio Controls provides information on how to use the different UI elements in the SoundStructure Studio software including knobs and matrix crosspoints.
- Appendix A: Command Protocol Reference Guide provides detailed information on the SoundStructure command protocol and the full command set.
- Appendix B: Address Book provides detailed information on how to use SoundStructure Studio's address book functionality to manage and connect to SoundStructure systems across an enterprise's network.
- Appendix C: Designing Audio Conferencing Systemsis an audio conferencing design guide. Refer to this section if new to audio conferencing or would like to better understand audio conferencing concepts.

If you are new to the SoundStructure products, read this guide starting with Introducing the Polycom SoundStructure Product Family for an overview, Customizing SoundStructure Designs to begin using SoundStructure Studio, and the remaining chapters as necessary to learn more about using SoundStructure products.

Introducing the Polycom SoundStructure Product Family

There are two product lines in the SoundStructure product family: the SoundStructure Conferencing series devices (C-series) designed for audio conferencing applications and the SoundStructure Sound Reinforcement series devices (SR-series) designed for commercial sound applications.

While the C-series and SR-series product families share a common design philosophy, both have audio processing capabilities that are designed for their respective applications. As described in detail below, the C-series products include acoustic echo cancellation on all inputs and are designed for audio and video conferencing applications. The SR-series products do not include acoustic echo cancellation and are designed for dedicated sound reinforcement, live sound, broadcast, and other commercial sound applications that do not require acoustic echo cancellation processing.

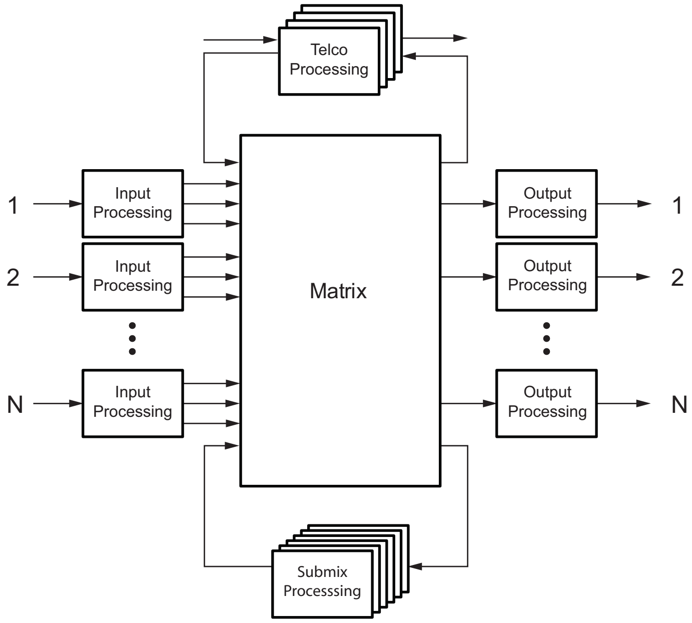
Defining SoundStructure Architectural Features

This section defines the common architectural features of the SoundStructure products and details the specific processing for both the C-series and SR-series products. Details on how to configure the devices are provided in Introducing SoundStructure Design Concepts, Creating Designs with SoundStructure Studio, and Customizing SoundStructure Designs.

All SoundStructure products are designed with the flexibility of an open architecture and the ease of design and installation of a fixed architecture system. The resulting solution is tremendous flexibility in how signals are processed while simultaneously making it easy to achieve exceptional system performance.

The SoundStructure processing includes input processing available on all inputs, output processing available on all outputs, submix processing available on all submix signals, telephony processing available on all optional telephony interfaces, and an audio matrix that connects this processing together. The high-level architecture is shown in the following figure for a SoundStructure device that has N inputs and N outputs. The specific input and output processing depends on the product family (C-series or SR-series) and is described later in this chapter.

SoundStructure High-Level Architecture


Telco Processing
1
2
N
Input Processing
Input Processing
Input Processing
Matrix
Output Processing
Output Processing
Output Processing
1
2
N
Submix Processsing

The following table summarizes the number of inputs, outputs, and submixes supported within each type of device. As shown in this table, each SoundStructure device has as many submixes as there are inputs to the device.

Supported SoundStructure Inputs, Outputs, and Submixes

	SoundStructure			
	C16	**C12**	**C8**	**SR12**
Inputs	**16**	**12**	**8**	**12**
Outputs	**16**	**12**	**8**	**12**
Submixes	**16**	**12**	**8**	**12**

A summary of the different types of processing in the C-series and SR-series products is shown in the following table. As can be seen in this table, the difference between the products is that the C-series products include acoustic echo cancellation while the SR-series products do not include acoustic echo cancellation. The processing capabilities are described in the following sections.

Types of C-series and SR-series Product Processing

Product Processing	**C-Series**	**SR-Series**
Input Processing		
Up to 8th order highpass and lowpass	4	4
1st or 2nd order high shelf and low shelf	4	4
10-band parametric equalization	4	4
Acoustic echo cancellation, 20-22kHz 200 msec tail-time, monaural or stereo	4	
Automatic gain control: +15 to -15dB	4	4

Types of C-series and SR-series Product Processing		
Dynamics processing: gate, expander, compressor, limiter, peak limiter	4	4
Feedback Eliminator: 10 adaptive filters	4	4
Noise cancellation: 0-20dB noise reduction	4	4
Automixer: gain sharing or gated mixer	4	4
Signal fader gain: +20 to -100 dB	4	4
Signal delay to 1000 msec	4	4
Output Processing		
1st or 2nd order high shelf and low shelf filters	4	4
10-bands of parametric or 31-band graphic equalizer	4	4
Dynamics processing: gate, expander, compressor, limiter, peak limiter	4	4
Signal fader gain: +20 to -100 dB	4	4
Cross over equalization up to 8th order highpass and lowpass filters, 1st order	4	4
Crossover delay: up to 100 msec	4	4
Signal delay: up to 1000 msec	4	4
Submix Processing		
Up to 8th order highpass and lowpass filters	4	4
1st or 2nd order high shelf and low shelf filters	4	4
10-bands of parametric equalization	4	4
Dynamics processing: gate, expander, compressor, limiter, peak limiter	4	4
Signal fader gain: +20 to -100 dB	4	4
Signal delay: up to 1000 msec	4	4
Telco Processing		
Line echo cancellation, 80-3300Hz, 32msec tail-time	4	4
Dynamics processing: gate, expander, compressor, limiter, peak limiter on telco transmit and receive	4	4
Up to 8th order highpass and lowpass filters	4	4
1st or 2nd order high shelf and low shelf filters	4	4
10-bands of parametric equalization on telco transmit and receive	4	4
Call progress detection	4	4
Signal fader gain: +20 to -100 dB	4	4
Automatic gain control: +15 to -15dB on telco receive	4	4
Signal delay on telco transmit and receive: up to 1000 msec	4	4
Noise cancellation: 0-20dB noise reduction on telco receive	4	4

Understanding Polycom OBAM™ - One Big Audio Matrix

One of the significant advancements in the SoundStructure products is the ability to link together multiple devices and configure and operate those devices as one large system rather than as multiple individual devices[1]. This feature dramatically simplifies any installation where audio from more than one device is required such as complicated sound reinforcement applications.

OBAM's 'one large system' approach provides many benefits including:

- Input signals that feed into the single matrix and outputs that are fed from the single matrix.

---

[1] Requires SoundStructure firmware release 1.2 or higher.

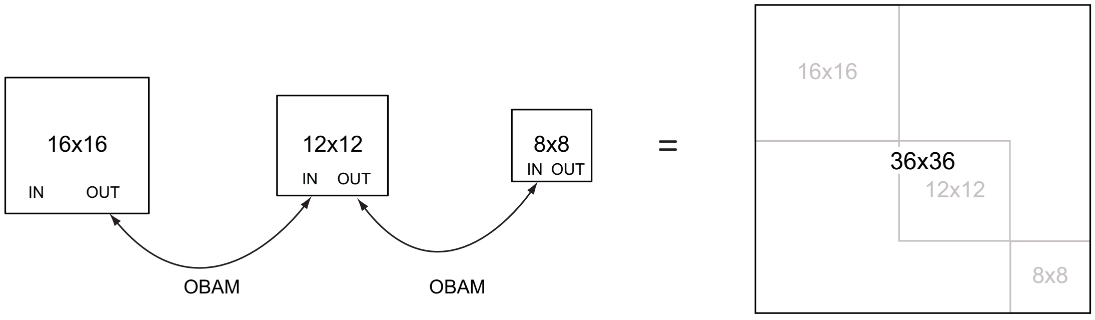
- No limitations on how signals from multiple devices are used together, which is beneficial for A/V designers.
- A transparent device linking scheme for all input signals that you share with all devices, which simplifies the setup, configuration, and maintenance of large systems.
- Inputs and outputs you can view on one screen, which eliminates the need to configure multiple devices by viewing multiple pages.

This one big system design approach is the result of the SoundStructure architectural design and the OBAM high-speed bi-directional link interface between devices. With OBAM, you can link up to eight devices together. If there are plug-in cards installed in multiple linked SoundStructure devices, the plug-in card resources are available for routing to any output across the system. See the Hardware Installation Guide or Introducing SoundStructure Design Concepts for more information on how to link multiple devices together.

The one large system design philosophy means that the audio matrix of a system of SoundStructure devices is the size of the total number of inputs and outputs of all the component devices that are linked together. Since one SoundStructure C16 device has a 16x16 matrix, two C16 devices linked together create a 32x32 matrix and so forth.

The OBAM architecture is shown in the following figure where a C16 device is linked to a C12 device which is linked to a C8 device. The resulting system has 36x36 inputs and 36 outputs (16+12+8 = 36). In addition to all the inputs and outputs, the submixes of each device also feed the matrix allowing the designer to have 36 submix signals (not shown in the following figure), one for each input that can be used in the system.

OBAM Architecture with Linked SoundStructure Devices

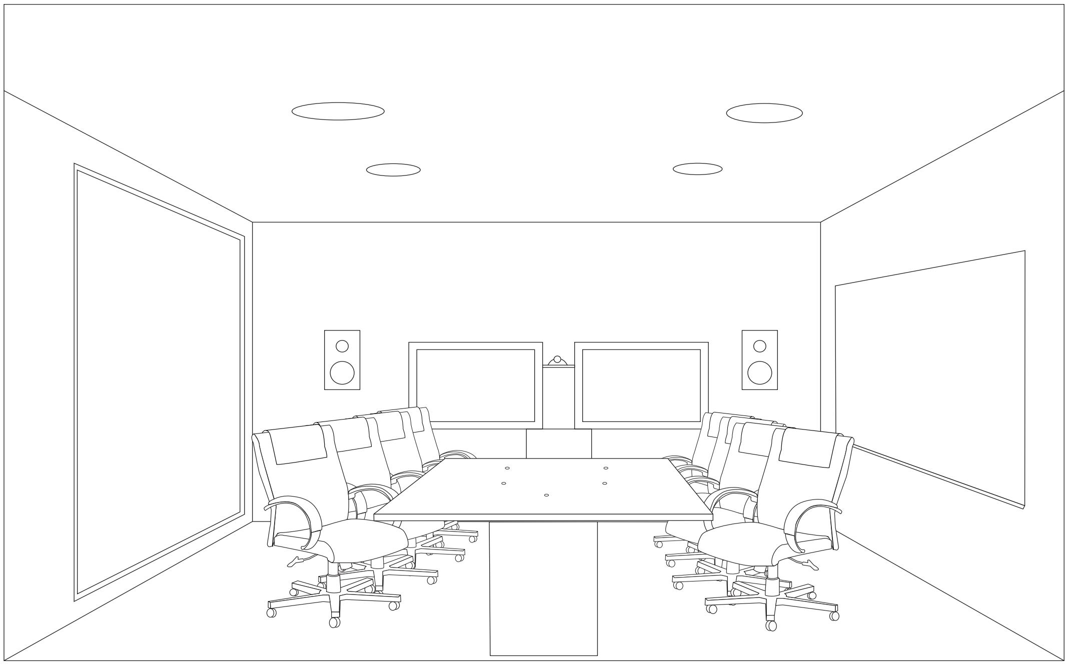


The OBAM design architecture helps A/V designers to no longer be concerned with device linking because multiple SoundStructure devices behave as, and are configured as, one large system.

Understanding SoundStructure C-Series Products

The SoundStructure C16, C12, and C8 devices are designed for audio conferencing applications where groups of people want to communicate to other individuals or groups such as in a typical room shown in the following figure.

A Conference Room Used with SoundStructure C-series Products



The SoundStructure C-series products feature both monaural and stereo acoustic echo cancellation, noise cancellation, equalization, dynamics processing, feedback elimination, and automatic microphone mixing.

Note: Processing Capability for Audio Inputs and Outputs

All audio inputs have the same processing capability and you can use audio inputs with either microphone-level or line-level inputs. Phantom power is available on all inputs.

All outputs have the same processing capability.

A single SoundStructure C16, C12, or C8 device supports 16, 12, or 8 microphone or line inputs and 16, 12, or 8 line outputs, respectively. You can link up to eight SoundStructure devices together including any combination of SoundStructure C-series or SR-series products may be used together to build audio processing systems that support up to 128 analog inputs and outputs.

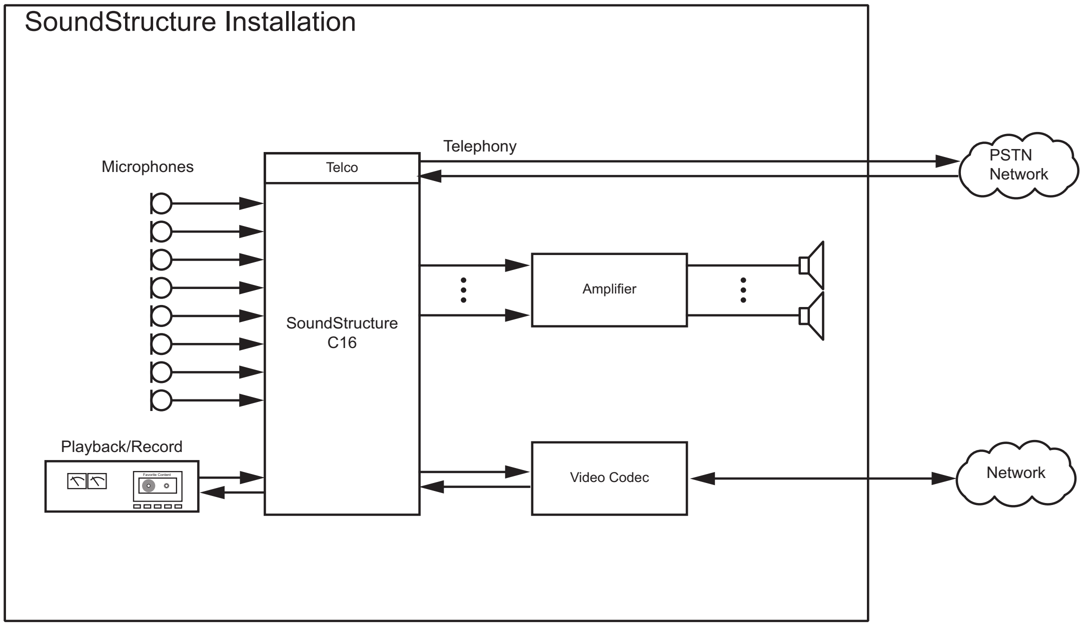
You can use each SoundStructure C-series device with traditional analog microphones or with Polycom's table and ceiling microphones[1]. For detailed information on using the Polycom table and ceiling microphones, see Connecting Over Conference Link2.

---

[1] Requires SoundStructure firmware release 1.1 or later.

Audio and video conferencing are typical applications of the SoundStructure C-series conferencing products where two or more remote locations are conferenced together. The typical connections in a conference room are shown in the following figure.

Typical SoundStructure Video and Audio Connections in a Conference Room


SoundStructure Installation
Microphones
Telco
Telephony
PSTN Network
SoundStructure C16
Amplifier
Playback/Record
Video Codec
Network

Before designing with SoundStructure products, the details of the SoundStructure signal processing capabilities are presented.

Understanding C-Series Input Processing

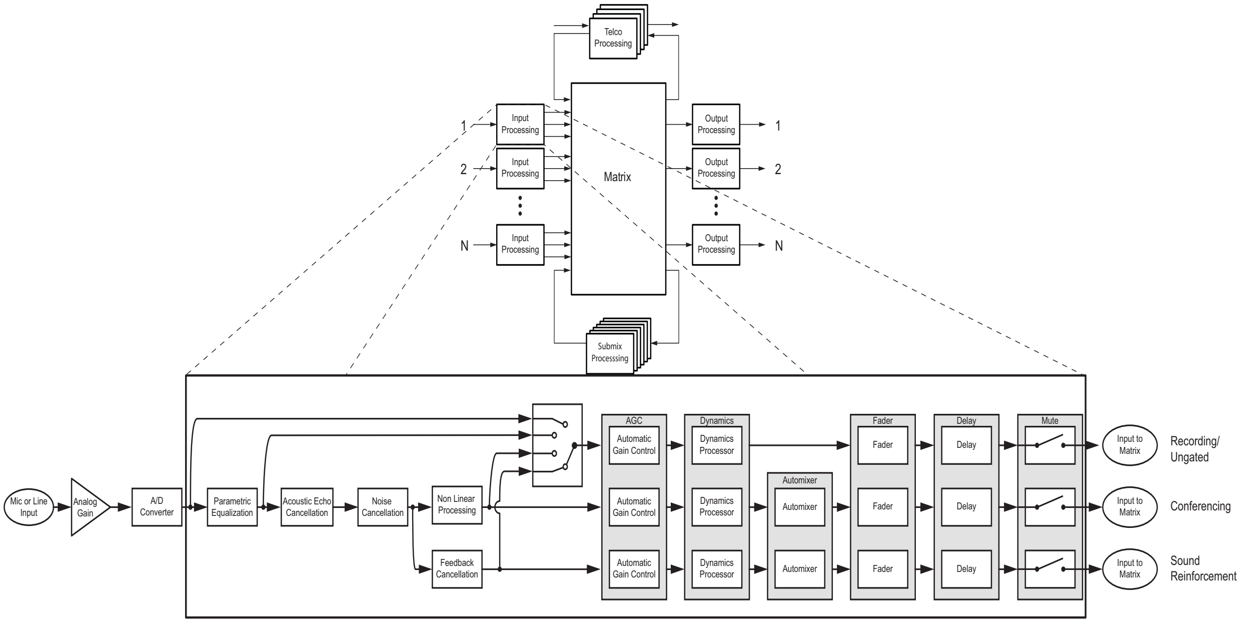
The input processing on the SoundStructure C-series devices is designed to help you create conferencing solutions with or without sound reinforcement. The audio input processing on a SoundStructure C-series device is shown in the following table.

SoundStructure Input Processing

Input Processing
Up to 8th order highpass and lowpass
1st or 2nd order high shelf and low shelf
10-band parametric equalization
Acoustic echo cancellation, 20-22kHz 200 msec tail-time, monaural or stereo
Automatic gain control: +15 to -15dB
Dynamics processing: gate, expander, compressor, limiter, peak limiter
Feedback Eliminator: 10 adaptive filters
Noise cancellation: 0-20dB noise reduction
Automixer: gain sharing or gated mixer
Signal fader gain: +20 to -100 dB
Signal delay to 1000 msec

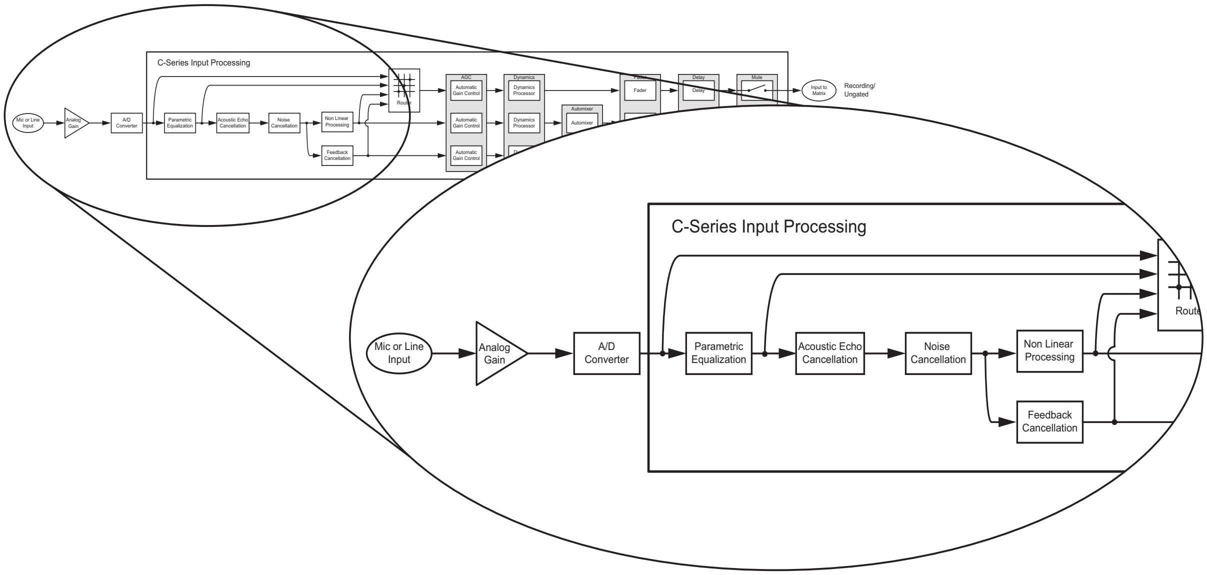
The signal processing follows the signal flow, as shown in the following figure.

SoundStructure C-Series Signal Processing and Signal Flow


Telco Processing
1
2
N
Input Processing
Input Processing
Input Processing
Matrix
Output Processing
Output Processing
Output Processing
1
2
N
Submix Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
AGC
Automatic Gain Control
Automatic Gain Control
Automatic Gain Control
Dynamics
Dynamics Processor
Dynamics Processor
Dynamics Processor
Automixer
Automixer
Automixer
Fader
Fader
Fader
Fader
Delay
Delay
Delay
Delay
Mute
Input to Matrix
Recording/ Ungated
Input to Matrix
Conferencing
Input to Matrix
Sound Reinforcement

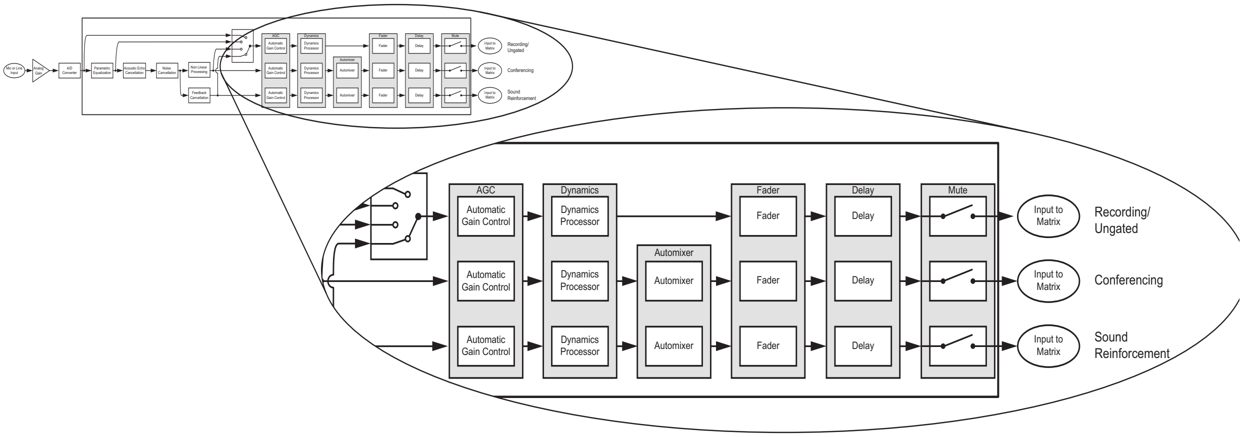
Each analog input signal has an analog gain stage that is used to adjust the gain of the input signal to the SoundStructure's nominal signal level of 0 dBu. The analog gain stage can provide from -20 to 64 dB of gain in 0.5 dB steps. There is also an option to enable 48 V phantom power on each input. Finally the analog input signal is digitized and available for processing. The digital signal is processed by five different DSP algorithms: parametric equalization, acoustic echo cancellation, noise cancellation, feedback reduction, and echo suppression (non linear processing).

SoundStructure C-Series Signal Input Processing


C-Series Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
Router
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated

Continuing through the signal path, as shown in the next figure, the input signal continues through the automatic gain control (AGC), dynamics processing, an automixer, an audio fader, and finally through the input delay.

SoundStructure C-Series Input Signal Path


AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement

Each analog input signal is processed to generate three different versions of the processed input signal that can be used simultaneously in the matrix:

- Conferencing version
- Sound reinforcement version
- Recording/ungated version

The AGC, dynamics processor, and input fader are linked together on all three audio paths, and each apply the same gain to the signal paths based on an analysis of the signal earlier in the signal path.

The automixer processing is applied to the conferencing and sound reinforcement signal paths to ensure that there is an un-automixed version of the input signal available for recording/ungated applications.

Note: Analog Input Signal Processing

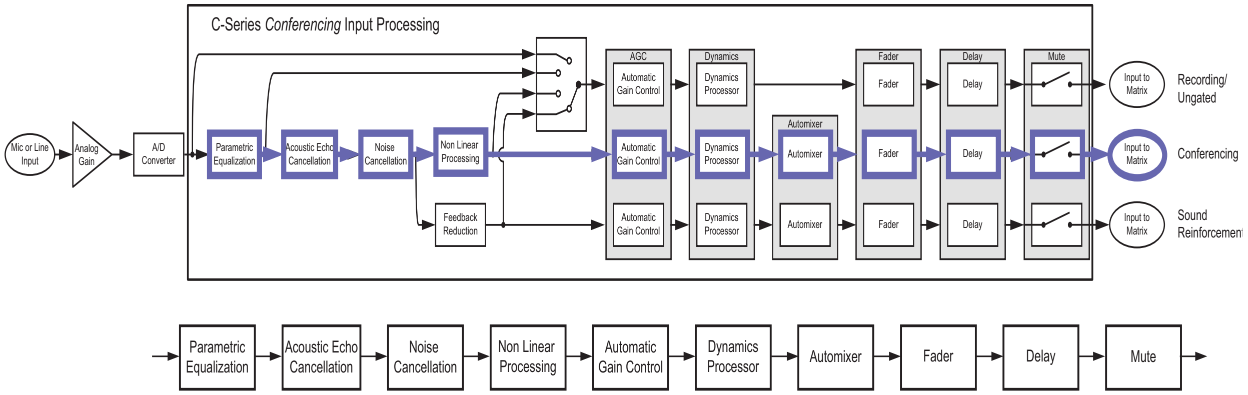
Each analog input signal is processed to create three processed versions that can be used in different ways in the matrix.

These three different versions of the input signal mean that, at the same time, an output signal to the loudspeakers can use the sound reinforcement processed version of an input signal, an output signal to the video conferencing system can use the conferencing processed version of the input signal, and an output signal to the recording system can use the recording processed version of the input signal. The decision of which of these three processed versions is used is made at each matrix crosspoint on the matrix as described in the Creating C-Series Matrix Crosspoints section below.

Processing Conferencing Version

The conferencing version is processed with the acoustic echo and noise cancellation settings, non-linear signal processing, automatic gain control, dynamics processing, automixer, fader, delay, and input mute. The conferencing signal path and summary block diagram is highlighted in the following figure. This is the path that is typically used to send echo and noise canceled microphone audio to remote locations. This is the default processing for microphone inputs when the automixed version of the signal is selected.

SoundStructure C-Series Conferencing Processing Signal Path

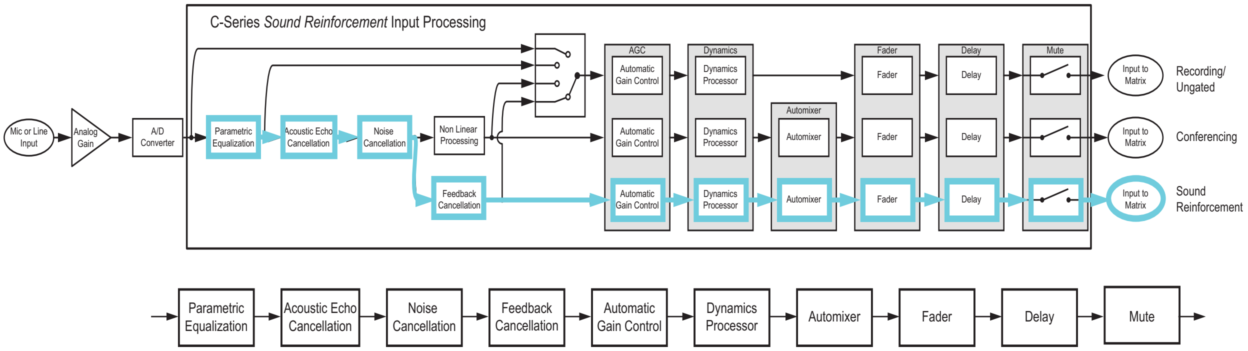

C-Series Conferencing Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Reduction
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Automatic Gain Control
Dynamics Processor
Automixer
Fader
Delay
Mute

Processing Sound Reinforcement Version

The sound reinforcement version is processed with the echo and noise cancellation, optional feedback elimination processing, automatic gain control, dynamics processing, automixer, fader, delay, and input mute. This is the path that is typically used for sending local audio to loudspeakers in the room for sound reinforcement. There is no non-linear processing on this path so that the local talker audio to the loudspeakers is not affected by the presence of remote talker audio in the local room.

The automatic gain control on the sound reinforcement path is different from the automatic gain control on the conferencing version of the signal because the sound reinforcement automatic gain control does not add gain to the signal. In other words, the sound reinforcement AGC only reduces the gain of the input signal. This restriction on the sound reinforcement AGC is to prevent the automatic gain control on the sound reinforcement path from increasing the microphone gain and consequently reducing the potential acoustic gain before the onset of feedback.

SoundStructure C-Series Sound Reinforcement Processing Signal Path


C-Series *Sound Reinforcement* Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
AGC
Automatic Gain Control
Automatic Gain Control
Automatic Gain Control
Dynamics
Dynamics Processor
Dynamics Processor
Dynamics Processor
Automixer
Automixer
Automixer
Fader
Fader
Fader
Fader
Delay
Delay
Delay
Delay
Mute
Input to Matrix
Recording/ Ungated
Input to Matrix
Conferencing
Input to Matrix
Sound Reinforcement
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Feedback Cancellation
Automatic Gain Control
Dynamics Processor
Automixer
Fader
Delay
Mute

Note: No Gain Control Added to Signal

The automatic gain control on the sound reinforcement processing path does not add gain to the signal, it only reduces the gain of the signal.

Processing Recording/Ungated Version

The recording version of the processed input signal is specifically designed to not include the gain sharing or gated style of automatic microphone mixing processing. The recording/ungated version of the input channel is typically used for recording applications or in any application where an un-automixed version of the input signal is required.

For additional flexibility in audio applications, there are four versions of the recording/ungated signal that you can select through the four-input router, as shown in the above processing figures. The selection of which type of recording/ungated signal to choose is performed on an input by input basis within the SoundStructure Studio software, as described in Customizing SoundStructure Designs.

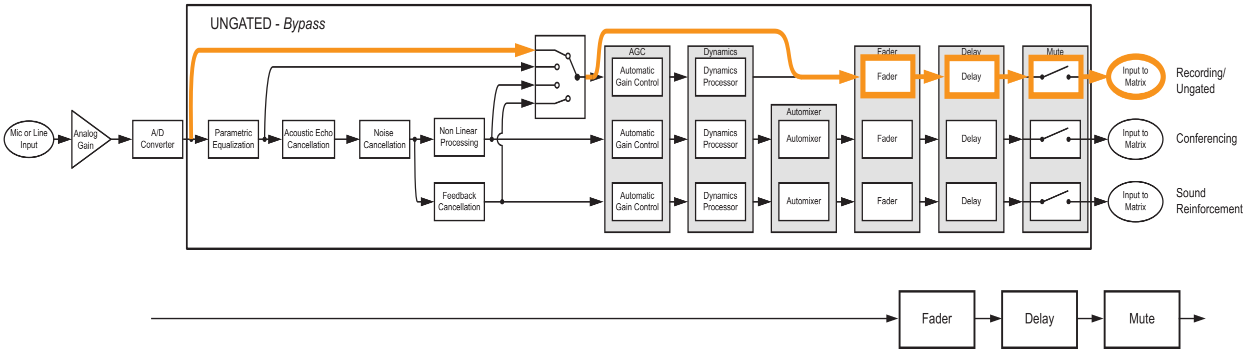
The four recording/ ungated versions are listed below:

- Bypass
- Line Input
- Conferencing
- Sound reinforcement

Processing Recording/Ungated–Bypass

The recording/ungated-bypass has no input processing other than a fader gain control, input delay, and input mute. This version bypasses the automatic gain control and dynamics processing, as shown in the following figure. You can use the bypass version for minimal audio processing on an input signal. This version of the signal has no acoustic echo cancellation processing and consequently includes any acoustic echo signal that may be present at the microphones.

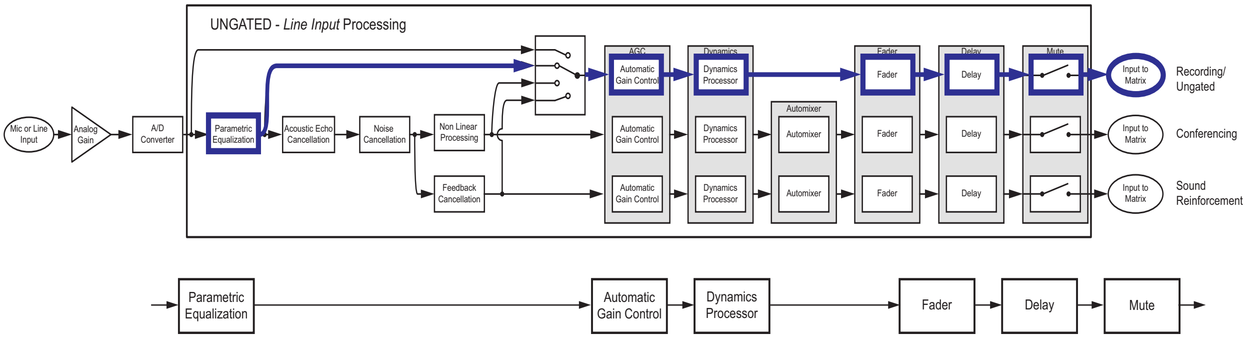
SoundStructure C-Series Recording/Ungated–Bypass Path


UNGATED - *Bypass*
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Fader
Delay
Mute

Processing Recording/Ungated–Line Input

The recording/ungated line input includes equalization, automatic gain control, and the dynamics processing as well as fader gain control, input delay, and input mute, as shown in the following figure. This processing path is typically used by line input signals such as program audio.

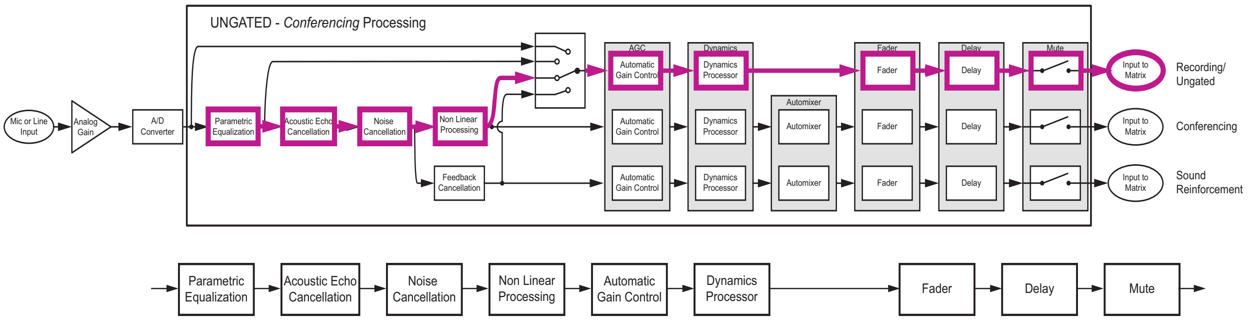
SoundStructure C-Series Recording Ungated–Line Input Path


UNGATED - *Line Input* Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Parametric Equalization
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

Processing Recording/Ungated–Conferencing

The recording/ungated conferencing processed input includes acoustic echo and noise cancellation, as shown in the following figure. This path is used for the recording of conference microphones as it includes all the acoustic echo cancellation but not the automatic microphone mixer processing.

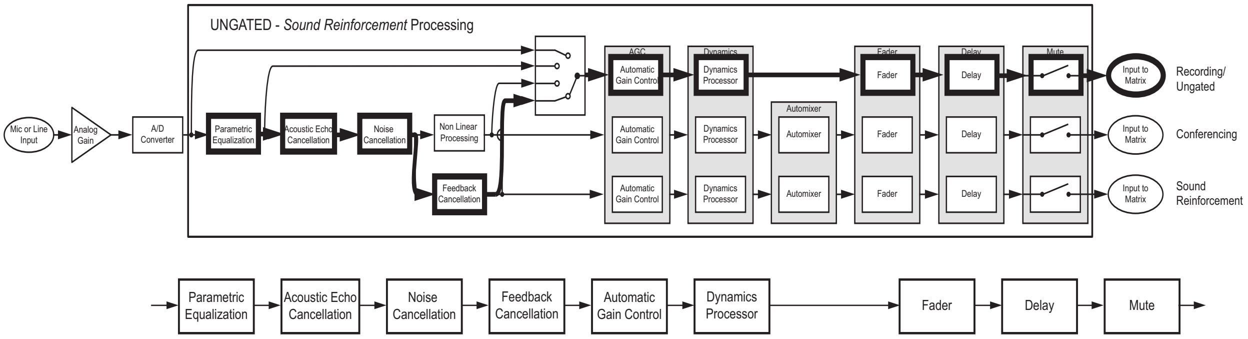
SoundStructure C-Series Recording Ungated–Conferencing Path


UNGATED - *Conferencing* Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

Processing Recording/Ungated–Sound Reinforcement

Finally, the sound reinforcement recording/ungated input includes the echo and noise cancellation and optional feedback elimination processing, as shown in the following figure.

SoundStructure C-Series Recording Ungated–Sound Reinforcement Path


UNGATED - *Sound Reinforcement* Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Feedback Cancellation
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

All versions of the recording/ungated input signal processing can be used simultaneously in the matrix. The conferencing version is typically used to send to remote participants, the sound reinforcement version is typically used to send to the local loudspeaker system, and the recording version is typically used for archiving the conference audio content.

Creating C-Series Matrix Crosspoints

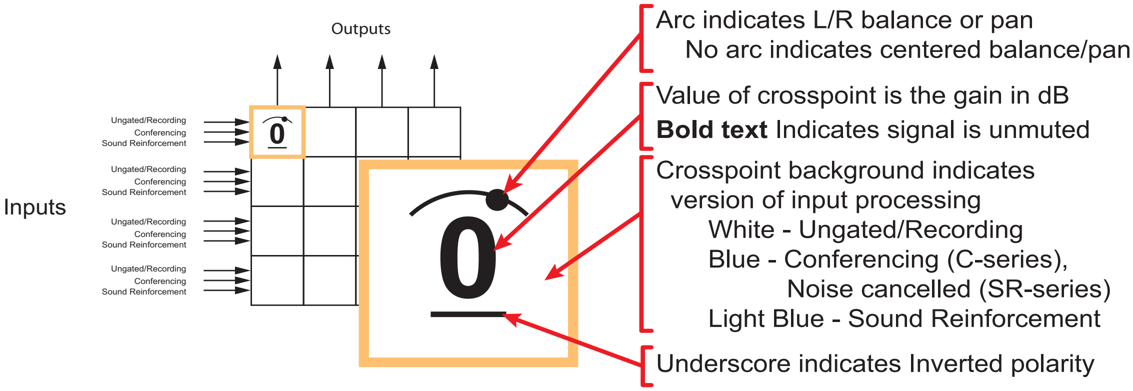
The audio matrix is used to create different mixes of input signals and submix signals are sent to output signals and submix signals. Matrix crosspoints gain values are shown in dB where 0 dB means a signal value is unchanged. For example, a crosspoint value of -6 dB lowers the signal gain by 6 dB before it is summed with other signals. You can adjust the matrix crosspoint gain in 0.1 dB steps between -100 and +20 dB, and you can also completely mute the matrix crosspoint. In addition, you can also negate and invert the matrix crosspoint so that the crosspoint arithmetic creates a subtraction rather than an addition. The inversion technique is effective in difficult room reinforcement environments by creating phase differences in alternating zones to add more gain before feedback.

Matrix crosspoints associated with stereo channels have a balance or pan to control mapping mono to stereo channels, stereo to mono channels, and stereo to stereo channels.

The three different versions of the input processing - the ungated, conferencing, and sound reinforcement - are selected at the matrix crosspoint. The SoundStructure Studio software allows the user to select which version of the input signal processing at the matrix crosspoint. As shown in Creating Designs with SoundStructure Studio, the different versions of the input processing are represented with different background colors in the matrix crosspoint.

The following figure highlights how to interpret the matrix crosspoints in the matrix.

SoundStructure C-Series Matrix Crosspoints


Outputs
Inputs
Ungated/Recording
Conferencing
Sound Reinforcement
Ungated/Recording
Conferencing
Sound Reinforcement
Ungated/Recording
Conferencing
Sound Reinforcement
Ungated/Recording
Conferencing
Sound Reinforcement
0
0

Arc indicates L/R balance or pan
No arc indicates centered balance/pan

Value of crosspoint is the gain in dB
Bold text Indicates signal is unmuted

Crosspoint background indicates version of input processing
White - Ungated/Recording
Blue - Conferencing (C-series), Noise cancelled (SR-series)
Light Blue - Sound Reinforcement

Underscore indicates Inverted polarity

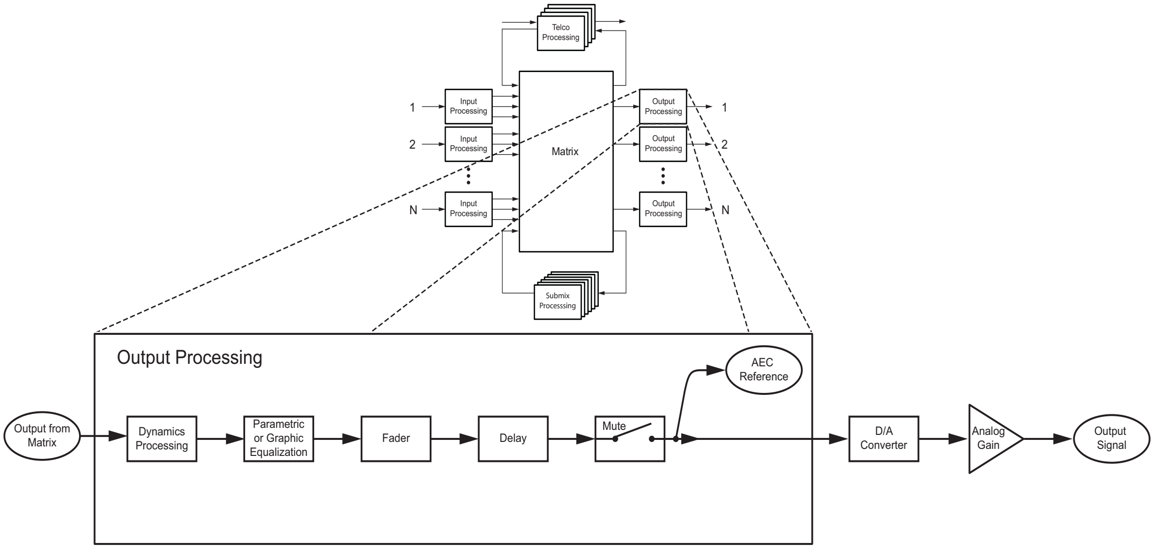
Understanding C-Series Output processing

As shown in the following table and figure, each output signal from the matrix can be processed with dynamics processing, either 10-band parametric or 10-, 15-, or 31-band graphic equalization, a fader, and output delay up to 1000 milliseconds.

SoundStructure C-Series Output Signal Processing

Output Processing
1st or 2nd order high shelf and low shelf filters
1st or 2nd order high shelf and low shelf filters
10-bands of parametric or 31-band graphic equalizer
Dynamics processing: gate, expander, compressor, limiter, peak limiter
Signal fader gain: +20 to -100 dB
Signal delay: up to 1000 msec

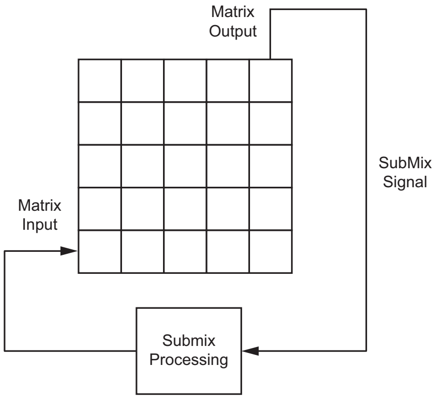
SoundStructure C-Series Output Signal Processing


Telco Processing
1
2
N
Input Processing
Input Processing
Input Processing
Matrix
Output Processing
Output Processing
Output Processing
1
2
N
Submix Processing
Output Processing
AEC Reference
Output from Matrix
Dynamics Processing
Parametric or Graphic Equalization
Fader
Delay
Mute
D/A Converter
Analog Gain
Output Signal

Processing C-Series Submixes

Submixes are outputs from the matrix that can be routed directly back to the input of the matrix as shown in the following figure.

SoundStructure C-Series Submix Signal Matrix


Matrix Output
SubMix Signal
Matrix Input
Submix Processing

As an output of the matrix, any combination of input signals can be mixed together to create the output submix signal. This output signal can be processed with the submix processing and the processed signal is available as an input to the matrix. Microphones, remote audio sources, or other signals are typically sent to a submix channel and the resulting submix signal is used as a single input in the matrix.

SoundStructure C-Series Submix Processing

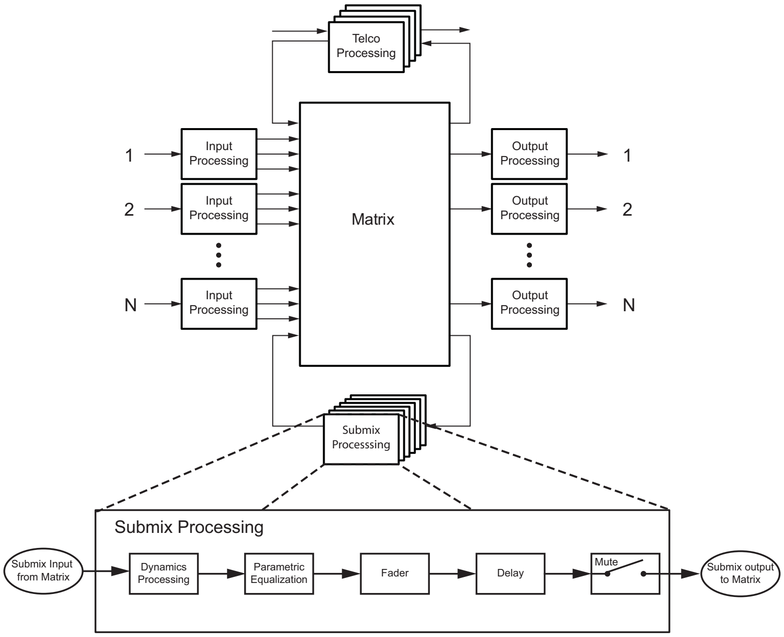
Submix Processing
Up to 8th order highpass and lowpass filters

SoundStructure C-Series Submix Processing

1st or 2nd order high shelf and low shelf filters
10-bands of parametric equalization
Dynamics processing: gate, expander, compressor, limiter, peak limiter
Signal fader gain: +20 to -100 dB
Signal delay: up to 1000 msec

As shown in the following figure, each submix signal from the matrix is processed with dynamics processing, parametric equalization, a fader, and up to 1000 milliseconds of delay. Each SoundStructure device has as many submixes as there are inputs.

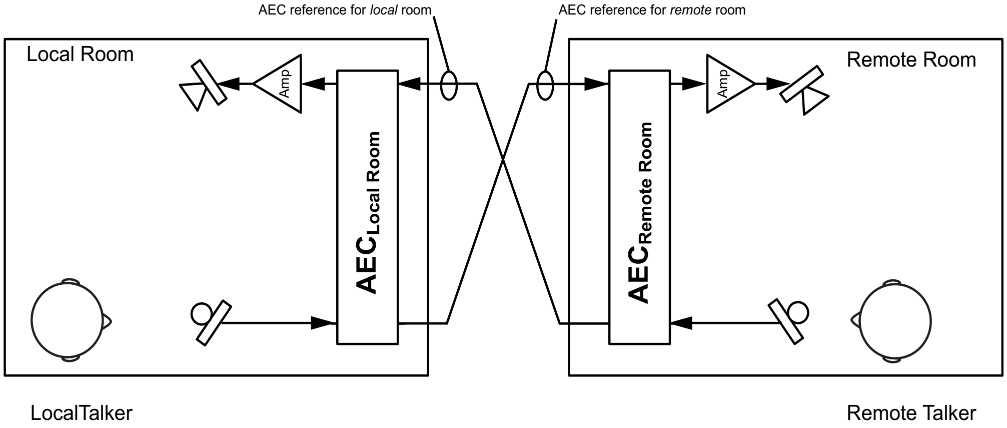
SoundStructure C-Series Submix Processing


Telco Processing
1
2
N
Input Processing
Input Processing
Input Processing
Matrix
Output Processing
Output Processing
Output Processing
1
2
N
Submix Processsing
Submix Processing
Submix Input from Matrix
Dynamics Processing
Parametric Equalization
Fader
Delay
Mute
Submix output to Matrix

Understanding C-Series Acoustic Echo Canceller References

In conferencing applications, an acoustic echo canceller (AEC) removes the remote site's audio that is played in the local room and prevents the audio from being picked up by the local microphones and sent back to the remote participants. The $\text{AEC}_{\text{Local Room}}$ in the following figure removes the acoustic echo of the remote talker so the audio is not sent back to the remote talker.

SoundStructure C-Series Acoustic Echo Cancellation Process



Acoustic echo cancellation processing is only required on the inputs that have microphone audio connected which can potentially hear both the local talkers' speech and the acoustic echo of the remote talkers' speech.

In order for the local acoustic echo canceller to cancel the acoustic echo of the remote participants, it must have an *echo canceller reference* defined. The echo canceller reference includes all the signals from the remote site that needs echo canceling. In the above figure, the AEC reference for both the local and remote rooms includes the audio that is played out the loudspeaker. See Appendix C: Designing Audio Conferencing Systems for additional information on audio conferencing systems and acoustic echo cancellation.

Within SoundStructure devices, the acoustic echo canceller on each input can have either one or two AEC references specified per input signal. For traditional monaural audio or video conferencing applications, only one acoustic echo canceller reference is used which is typically sent to the single loudspeaker zone. See the Creating an Eight Microphones, Video, and Telephony Application Conferencing System in Creating Advanced Applications for an example.

Applications that have two independent audio sources played into the room such as stereo audio from a stereo video codec require two mono AEC references, or one stereo AEC reference. See Creating an Eight Microphones and Stereo Video Conferencing System in Creating Advanced Applications.

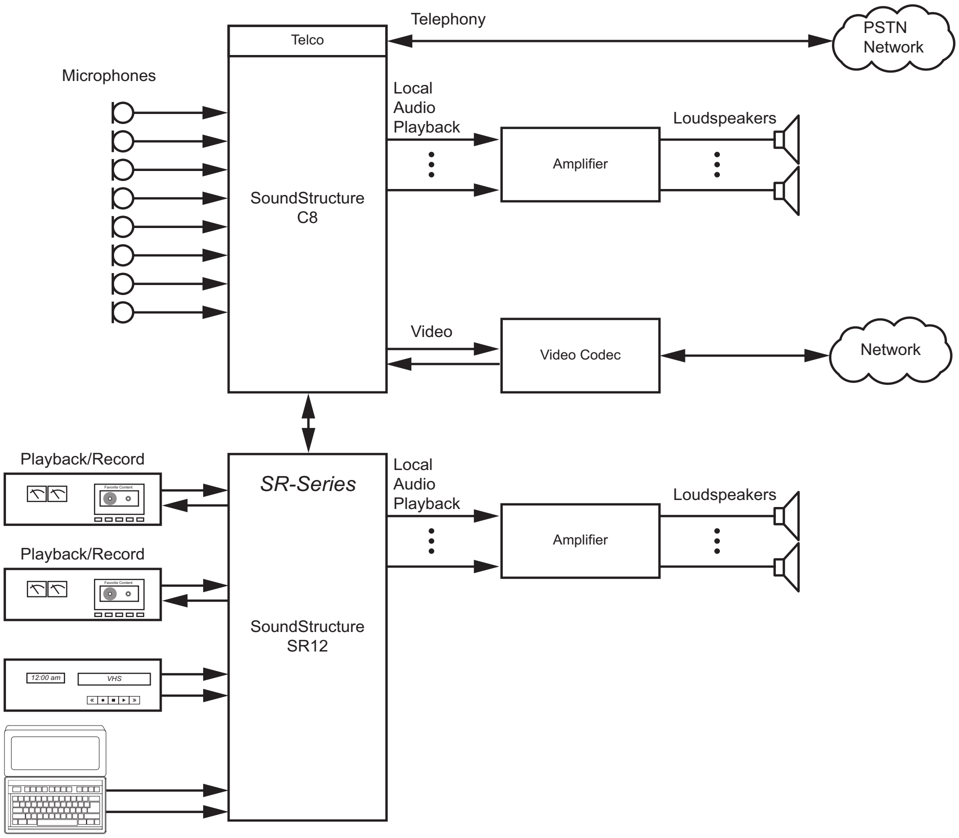
You can create an acoustic echo canceller reference from any output signal or any submix signal. For a SoundStructure C16 device, this means that there are 32 possible echo canceller references (16 outputs + 16 submixes) that you can define and select.

Understanding SoundStructure SR-Series Products

The SoundStructure SR12 has a similar architecture to the SoundStructure C-series. While the SoundStructure SR12 does not include acoustic echo cancellation processing, the SR12 does include noise cancellation, automatic microphone mixing, matrix mixing, equalization, feedback elimination, dynamics processing, delay, and submix processing.

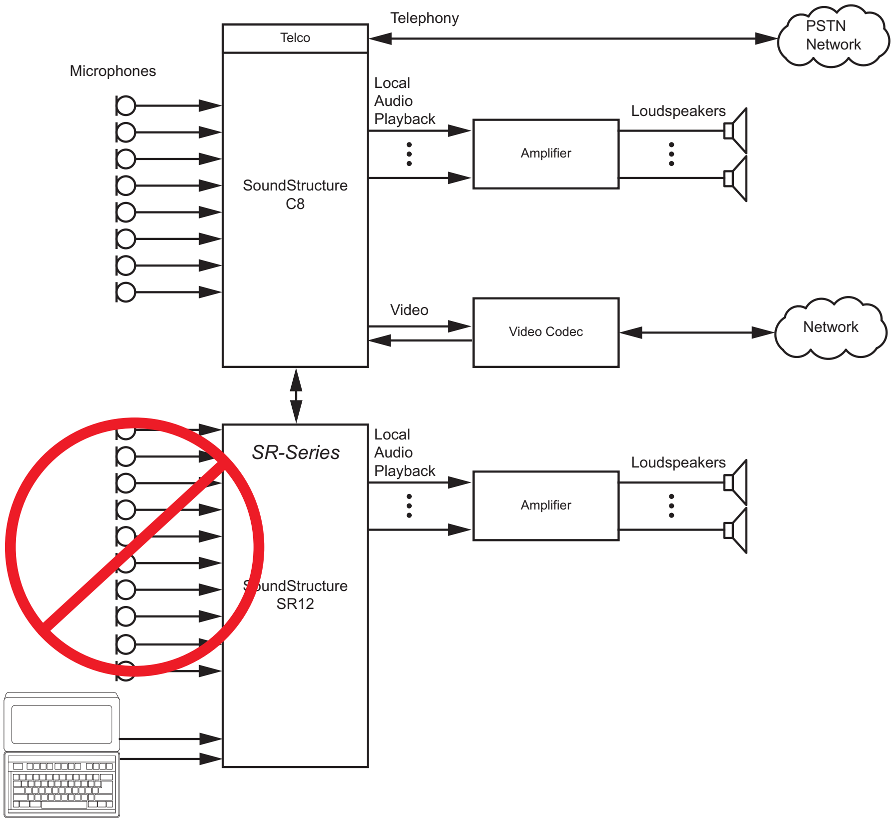
The SoundStructure SR12 is designed for both the non-conferencing applications where local audio is played into the local room or distributed throughout a facility and for conferencing applications to provide additional line input and output signals when linked to a C-series product. Applications for the SoundStructure SR12 include live sound, presentation audio, sound reinforcement, and broadcasting. The following figure shows an example of using the SoundStructure SR12 to provide additional line level inputs and outputs to a SoundStructure C8 conferencing product.

SoundStructure SR12 Providing Line Level Inputs and Outputs for a SoundStructure C8


Telco
Telephony
PSTN Network
Microphones
Local Audio Playback
Loudspeakers
Amplifier
SoundStructure C8
Video
Video Codec
Network
Playback/Record
SR-Series
Local Audio Playback
Loudspeakers
Amplifier
Playback/Record
SoundStructure SR12
12:00 am
VHS

The SoundStructure SR12 can not be used to add additional conferencing microphones to a C series product because there is no acoustic echo cancellation processing on the SoundStructure SR12 inputs. The following figure shows an installation that does not work because the microphones that are connected to the SoundStructure SR12 are not echo canceled. If you need more conferencing microphones than can be used with a particular SoundStructure C-series device, you can use either the next largest C-series device or additional C-series devices to support the number of microphones required.

Installation Not Supported with SoundStructure SR12


Telephony
Telco
PSTN Network
Microphones
Local Audio Playback
Loudspeakers
Amplifier
SoundStructure C8
Video
Video Codec
Network
SR-Series
Local Audio Playback
Loudspeakers
Amplifier
SoundStructure SR12

You can use the C-series and SR-series products together and link the devices to form larger systems that can support up to eight SoundStructure devices, 128 inputs, 128 outputs, and eight plug-in daughter cards.

For information on how to rack mount and terminate cables to the SoundStructure devices, refer to the SoundStructure Hardware Installation Guide.

Understanding SR-Series Input Processing

The input processing on the SoundStructure SR-series devices is designed to make it easy to create commercial sound and sound reinforcement solutions. Each audio input on a SoundStructure SR-series device includes the signal processing path shown in the following table.

SoundStructure SR-Series Signal Input Processing Path

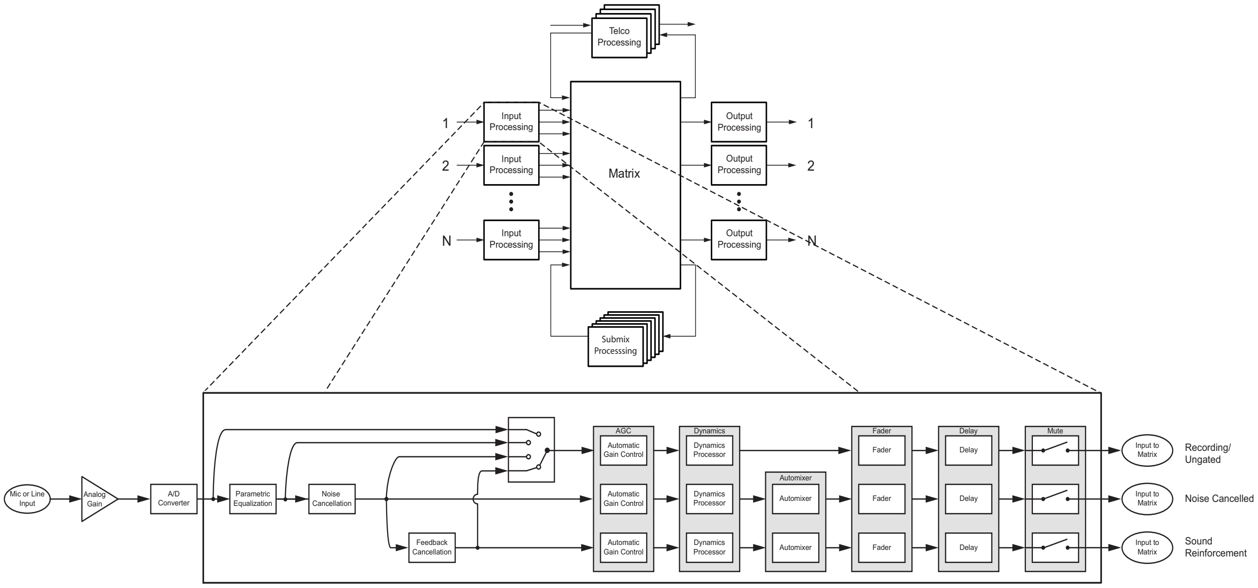
SR-Series Input Processing
Up to 8th order highpass and lowpass
1st or 2nd order high shelf and low shelf
10-bands of parametric equalization
Automatic gain control: +15 to -15dB
Dynamics processing: gate, expander, compressor, limiter, peak limiter

SoundStructure SR-Series Signal Input Processing Path

- Feedback Eliminator: 10 adaptive filters
- Noise cancellation: 0-20dB noise reduction
- **Automixer: gain sharing or gated mixer**
  - Signal fader gain: +20 to -100 dB
  - Signal delay: up to 1000 msec

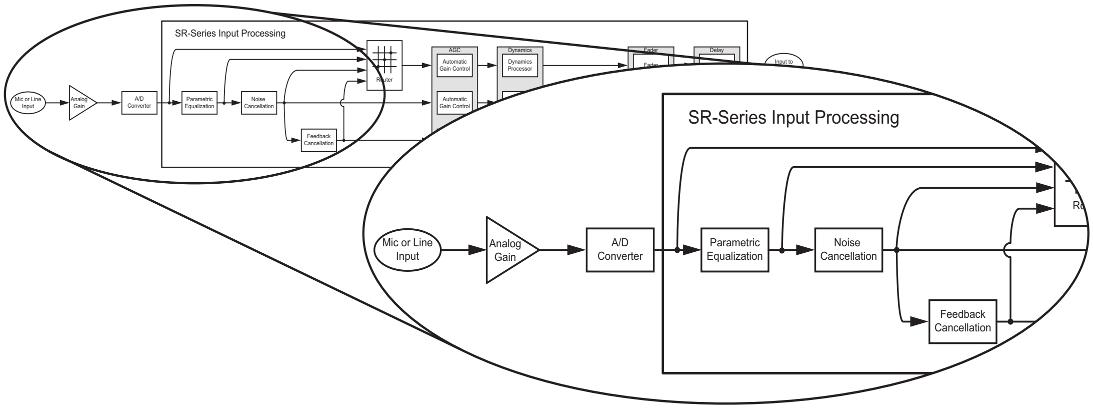
The processing for each input is shown in the following figure from analog input signal to the three versions of input processing that lead to the matrix.

SoundStructure SR-Series Input Processing


Telco Processing
1
2
N
Input Processing
Input Processing
Input Processing
Matrix
Output Processing
Output Processing
Output Processing
1
2
N
Submix Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Noise Cancellation
Feedback Cancellation
AGC
Automatic Gain Control
Automatic Gain Control
Automatic Gain Control
Dynamics
Dynamics Processor
Dynamics Processor
Dynamics Processor
Automixer
Automixer
Automixer
Fader
Fader
Fader
Fader
Delay
Delay
Delay
Delay
Mute
Input to Matrix
Input to Matrix
Input to Matrix
Recording/ Ungated
Noise Cancelled
Sound Reinforcement

Each analog input signal has an analog gain stage that is used to adjust the gain of the input signal to the SoundStructure's nominal signal level of 0 dBu. The analog gain stage can provide from -20 to 64 dB of analog gain in 0.5 dB increments. There is also an option to enable 48 V phantom power on each input. Finally, the analog input signal is digitized and ready for processing.

SoundStructure SR-Series Input Processing


SR-Series Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Noise Cancellation
Feedback Cancellation
Router
AGC
Automatic Gain Control
Automatic Gain Control
Dynamics
Dynamics Processor
Fader
Fader
Delay
Input to
SR-Series Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Noise Cancellation
Feedback Cancellation

Continuing through the signal path as shown in the next figure, the input signal processing continues through the automatic gain control (AGC), dynamics processing, an automixer, an audio fader, and finally through the input delay.

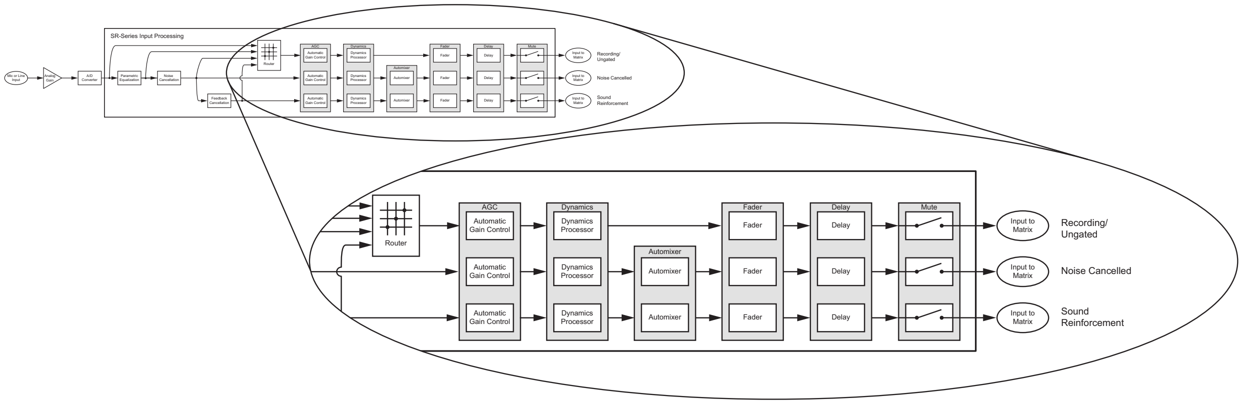
Each analog input signal is processed to generate three different versions of the processed input signal that can be used simultaneously in the matrix. The following are the three versions of processed input signal:

- Noise canceled
- Sound reinforcement
- Recording/ungated

The AGC, dynamics processor, and input fader are linked together on all three audio paths and apply the same gain to the signal paths based on an analysis of the signal earlier in the signal path.

The automixer processing is only applied to the noise canceled and sound reinforcement signal paths to ensure that there is an un-automixed version of the input signal available for recording/ungated applications.

SoundStructure SR-Series Processed Input Signals


SR-Series Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Noise Cancellation
Feedback Cancellation
Router
AGC
Automatic Gain Control
Automatic Gain Control
Automatic Gain Control
Dynamics
Dynamics Processor
Dynamics Processor
Dynamics Processor
Automixer
Automixer
Automixer
Fader
Fader
Fader
Fader
Delay
Delay
Delay
Delay
Mute
Input to Matrix
Recording/ Ungated
Input to Matrix
Noise Cancelled
Input to Matrix
Sound Reinforcement

Note: Analog Input Signal Processing

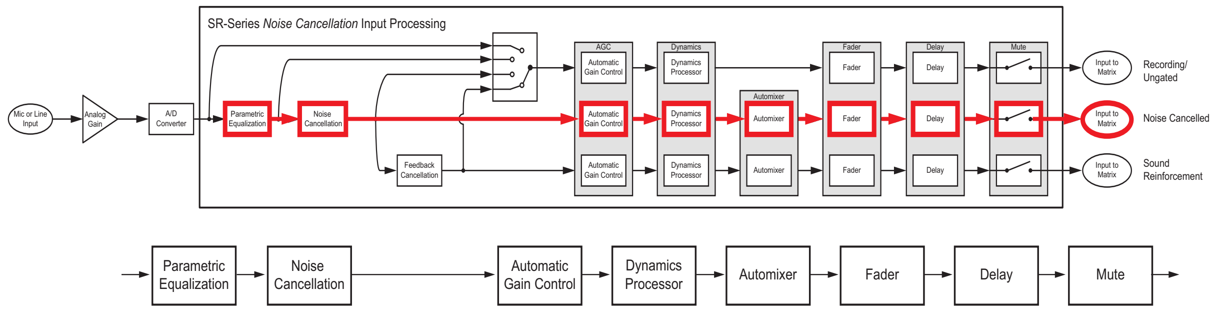
Each analog input signal is processed to create three processed versions that are used in different ways in the matrix.

These three different versions of the input signal mean that, at the same time, an output signal to the loudspeakers can use the sound reinforcement processed version of an input signal, another output signal can use the noise canceled version without feedback processing, and a different output signal can use the recording version of the input signal. The decision of which of these three processed versions to use is made at each matrix crosspoint as described in Creating SR-Series Matrix Crosspoints.

Processing Noise Canceled

The conferencing version is processed with input equalization, noise cancellation, automatic gain control, dynamics processing, automixer, fader, delay, and input mute. The noise canceled signal path is highlighted in the following figure and the block diagram of this processing is also shown. This is the path that is typically used to send a noise reduced version of the microphone audio to paging zones that are not acoustically coupled to the microphone. This is the default processing for microphone inputs when the automixed version of the signal is selected.

SoundStructure SR-Series Noise Cancellation Processing

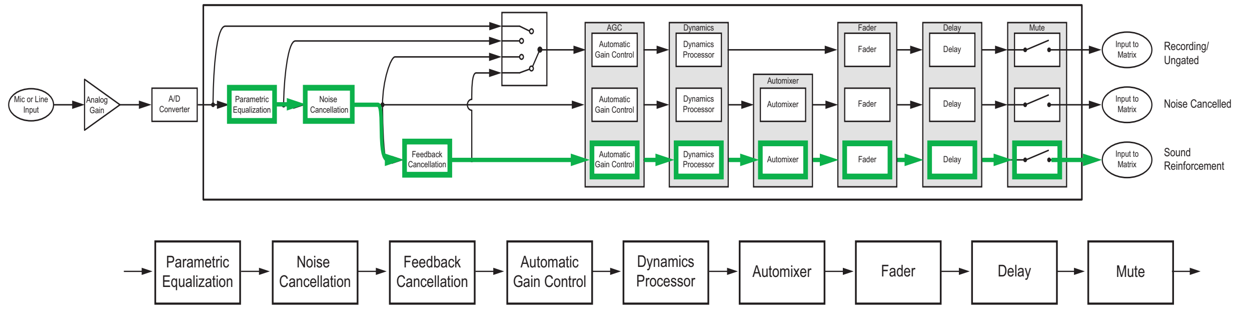

SR-Series Noise Cancellation Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Noise Cancellation
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Noise Cancelled
Sound Reinforcement
Parametric Equalization
Noise Cancellation
Automatic Gain Control
Dynamics Processor
Automixer
Fader
Delay
Mute

Processing Sound Reinforcement

The sound reinforcement version is processed with the parametric equalization, noise cancellation, optional feedback elimination processing, automatic gain control, dynamics processing, automixer, fader, delay, and input mute. This is the path that is typically used for sending local audio to loudspeakers in the room for sound reinforcement.

The automatic gain control on the sound reinforcement path is different from the automatic gain control on the noise canceled version of the signal in that the sound reinforcement automatic gain control does not add gain to the signal. In other words, the sound reinforcement AGC only reduces the gain of the signal and does not add gain to the signal. This restriction on the sound reinforcement AGC prevents the automatic gain control from reducing the available potential acoustic gain before the onset of feedback.

SoundStructure SR-Series Sound Reinforcement Input Processing


Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Noise Cancellation
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Noise Cancelled
Sound Reinforcement
Parametric Equalization
Noise Cancellation
Feedback Cancellation
Automatic Gain Control
Dynamics Processor
Automixer
Fader
Delay
Mute

Processing Recording/Ungated Version

The recording version of the processed input signal is specifically designed to not include any gain sharing or gated-style of automatic microphone mixing processing. The recording/ungated version of the input is used for recording applications or in any application where an un-automixed version of the input signal is required.

For additional flexibility in audio applications, there are four different versions of the recording/ungated signal that can be selected through the four-input router shown in the previous processing figures. This selection of which type of recording/ungated signal to choose is performed on an input by input basis within the SoundStructure Studio software as described in Customizing SoundStructure Designs.

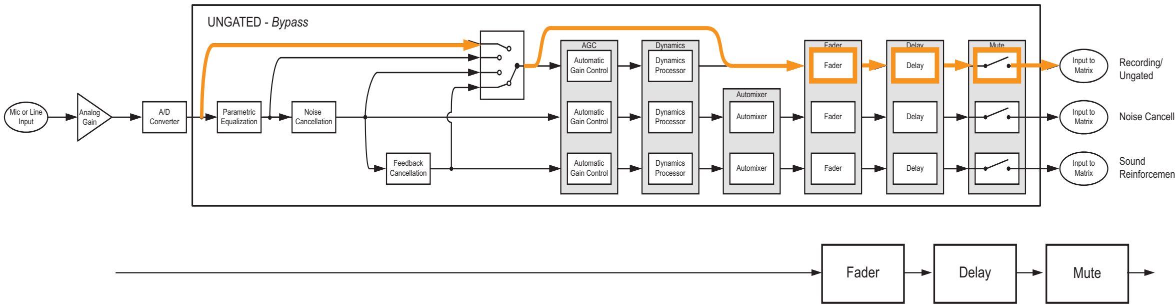
The following are four ungated versions of the processed input signal:

- Bypass
- Line input
- Noise cancellation
- Sound reinforcement

Processing Recording/Ungated–Bypass

The recording/ungated bypass version has no input processing other than a fader gain control, input delay, and input mute. This version bypasses the automatic gain control and dynamics processing, as shown in the following figure. This version can be used when it is important to have minimal audio processing on an input signal.

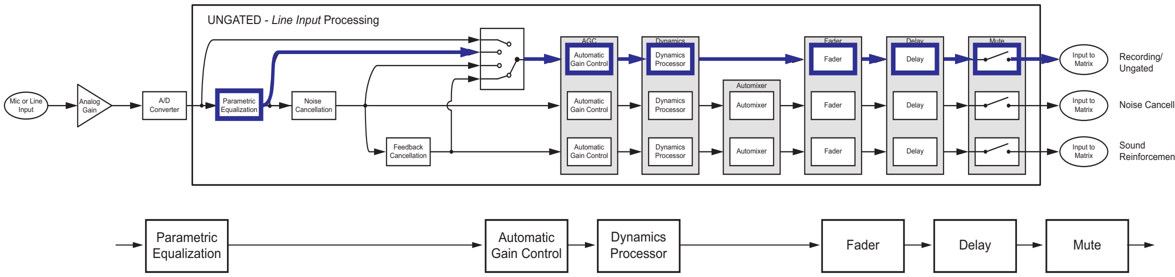
SoundStructure SR-Series Bypass Signal Processing


UNGATED - Bypass
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Noise Cancellation
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Noise Cancelled
Sound Reinforcement
Fader
Delay
Mute

Processing Recording/Ungated–Line Input

The recording line input version includes equalization, automatic gain control, and the dynamics processing as well as fader gain control, input delay, and input mute, as shown in the next figure. This processing path is typically used by line input signals such as program audio, and hence the name *line input* path.

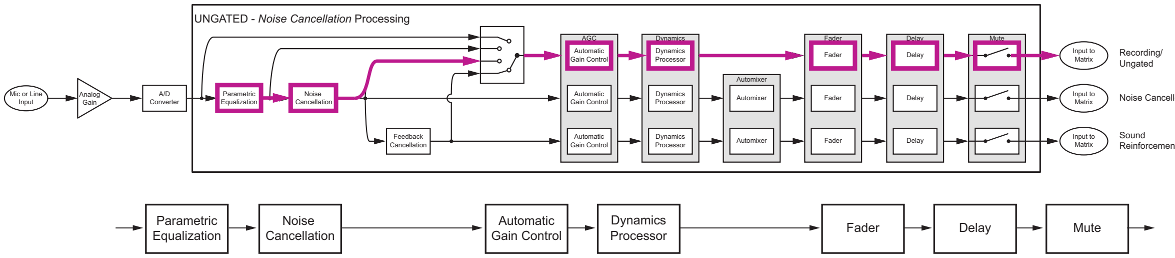
SoundStructure SR-Series Line Input Signal Processing


UNGATED - Line Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Noise Cancellation
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Noise Cancelled
Sound Reinforcement
Parametric Equalization
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

Processing Recording/Ungated - Noise Cancellation

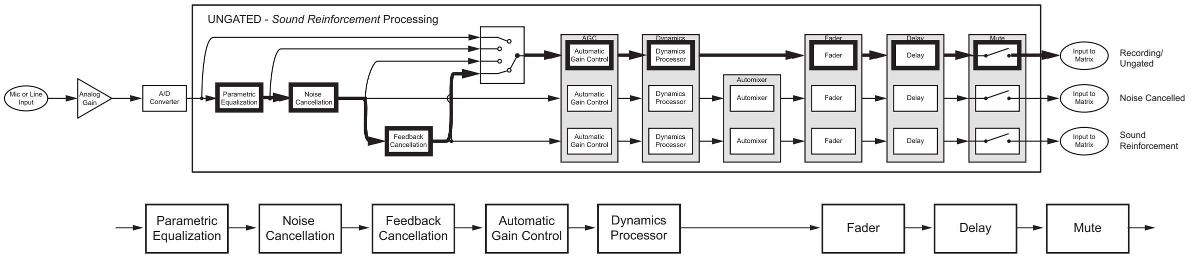
The noise canceled recording input includes the noise cancellation as shown in the next figure. This path is typically used for recording of microphone audio as it includes all the noise cancellation but not the automatic microphone mixer processing.

SoundStructure SR-Series Noise Cancellation Signal Processing


UNGATED - Noise Cancellation Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Noise Cancellation
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Noise Cancelled
Sound Reinforcement
Parametric Equalization
Noise Cancellation
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

Processing Recording/Ungated - Sound Reinforcement

Finally, the sound reinforcement recording input includes the noise cancellation and optional feedback elimination processing as shown in the following figure.


UNGATED - *Sound Reinforcement* Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Noise Cancellation
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Noise Cancelled
Sound Reinforcement
Parametric Equalization
Noise Cancellation
Feedback Cancellation
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

Creating SR-Series Matrix Crosspoints

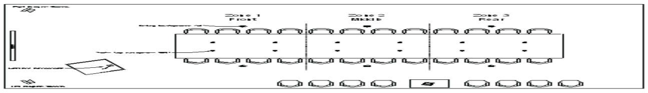
The audio matrix is used to create different mixes of input signals and submix signals to be sent to output signals and submix signals. Matrix crosspoints gain values are shown in dB where 0 dB means that the signal level is unchanged. Matrix crosspoint gains can be adjusted in 0.1 dB steps between -100 and +20 dB and may also be completely muted. In addition, the matrix crosspoint can also be negated/inverted so that the crosspoint arithmetic creates a subtraction instead of an addition.

Matrix crosspoints associated with stereo virtual channels have a balance or pan to control mapping mono to stereo virtual channels, stereo to mono virtual channels, and stereo to stereo virtual channels.

The different versions of the input processing are selected at the matrix crosspoint. The user interface provides an option for selecting the different versions of the input processing including the noise canceled, sound reinforcement, and ungated/recording version. As shown in Creating Designs with SoundStructure Studio, different versions of the input processing are represented with different background colors at the matrix crosspoint. The SoundStructure Studio software allows the user to select which version of the input signal processing at the matrix crosspoint.

The next figure shows how to interpret the matrix crosspoint view.

SoundStructure SR-Series Matrix Crosspoint



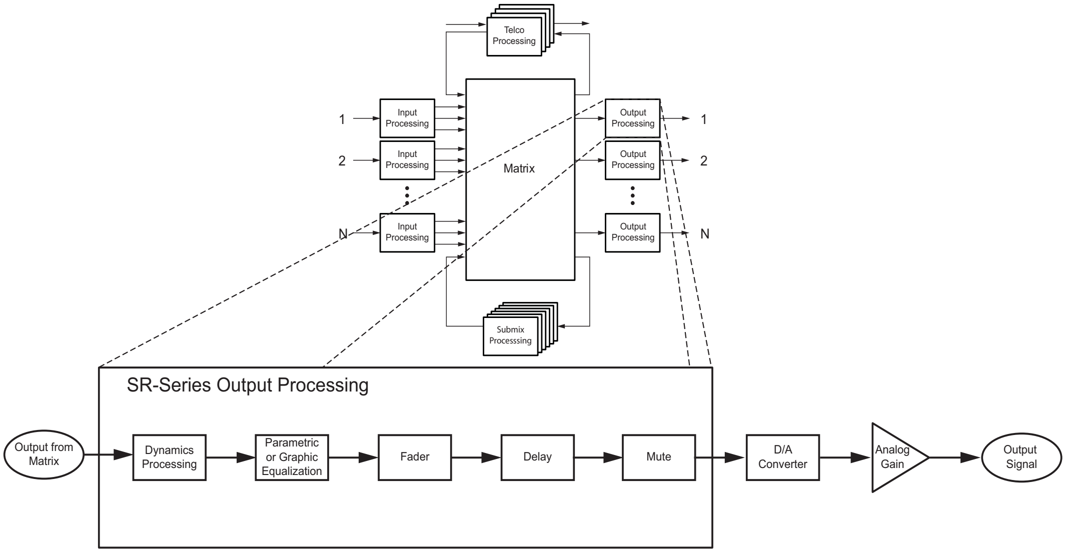
Understanding SR-Series Output Processing

The output processing for the SR-series of products is identical to the processing for the output processing in the C-series and is shown in the table and following figure.

SoundStructure SR-Series Output Processing

Output Processing
1st or 2nd order high shelf and low shelf filters
10-bands of parametric or 31-band graphic equalizer
Dynamics processing: gate, expander, compressor, limiter, peak limiter
Signal fader gain: +20 to -100 dB
Signal delay: up to 1000 msec

SoundStructure SR-Series Output Processing


Telco Processing
1
Input Processing
2
Input Processing
N
Input Processing
Matrix
Output Processing
1
Output Processing
2
Output Processing
N
Submix Processsing
SR-Series Output Processing
Output from Matrix
Dynamics Processing
Parametric or Graphic Equalization
Fader
Delay
Mute
D/A Converter
Analog Gain
Output Signal

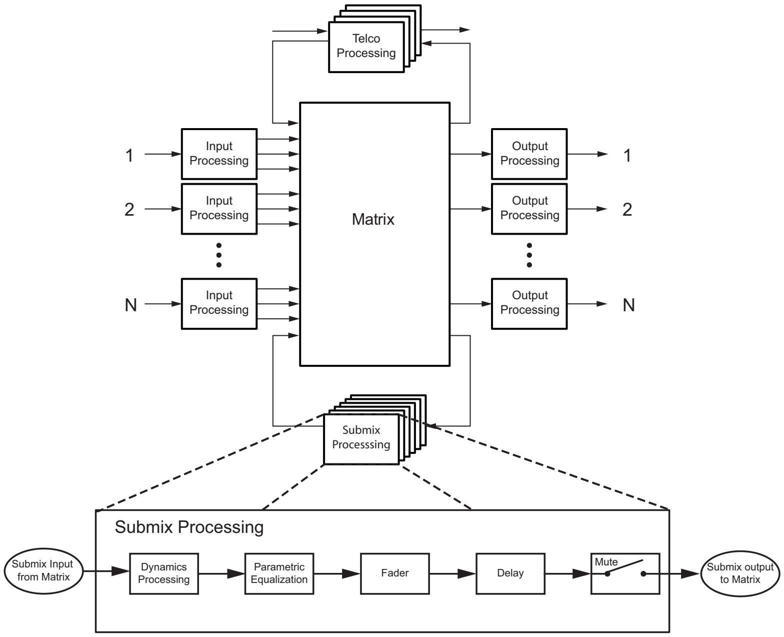
Processing SR-Series Submix

The submix processing for the SR-series of products is identical to the processing for the submix processing in the C-series and shown in the following table and figure.

SoundStructure SR-Series Submix Processing

Submix Processing
Up to 8th order highpass and lowpass filters
1st or 2nd order high shelf and low shelf filters
10-bands of parametric equalization
Dynamics processing: gate, expander, compressor, limiter, peak limiter
Signal fader gain: +20 to -100 dB
Signal delay: up to 1000 msec

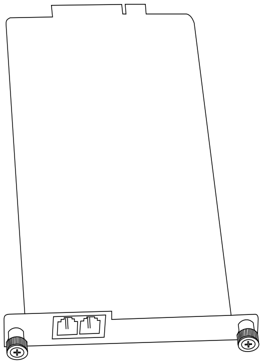
SoundStructure SR-Series Submix Processing


Telco Processing
1
2
N
Input Processing
Input Processing
Input Processing
Matrix
Output Processing
Output Processing
Output Processing
1
2
N
Submix Processsing
Submix Processing
Submix Input from Matrix
Dynamics Processing
Parametric Equalization
Fader
Delay
Mute
Submix output to Matrix

Understanding Telephony Processing

Both the C-series and SR-series SoundStructure devices support optional plug-in cards. Currently there are two telephony cards: TEL1, a single-PSTN line, and TEL2, a dual-PSTN line interface card in the form factor, shown in the following figure.

SoundStructure Telephony Card



These cards are field-installable and are ordered separately from the SoundStructure C- or SR-series devices. See the SoundStructure Hardware Installation Guide or the Hardware Installation Guide for the TEL1 and TEL2 for additional information.

The SoundStructure telephony cards have been designed to meet various regional telephony requirements through the selection of a country code from the user interface. For each telephony interface card, the signal processing is listed in the following table and shown in the following figure.

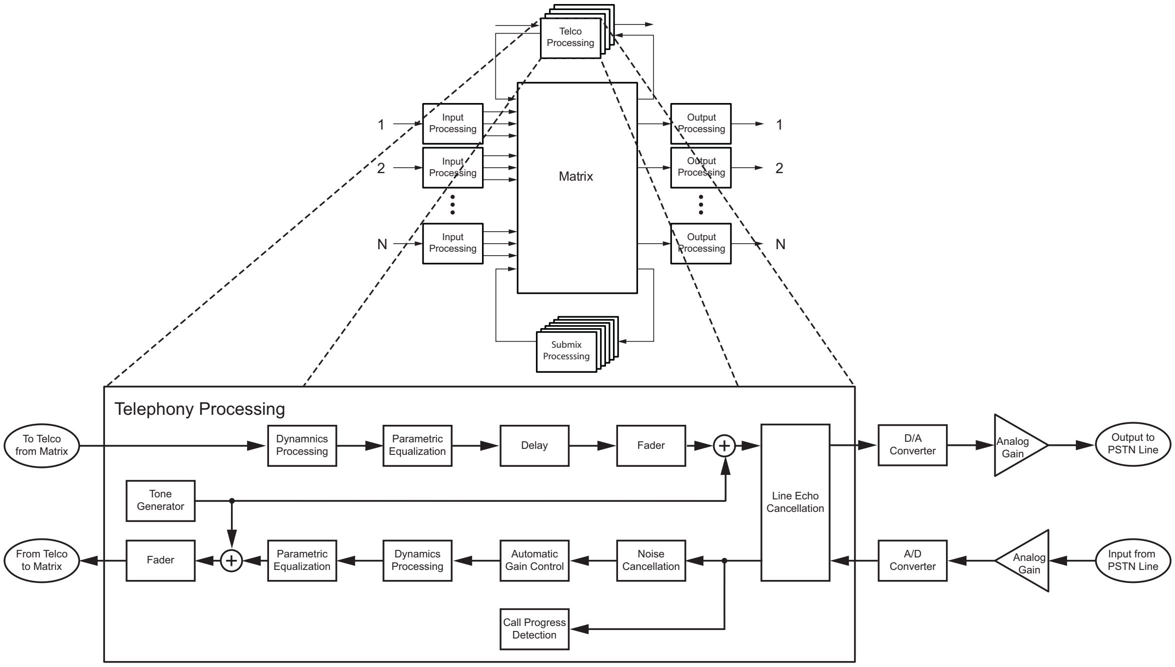
The telephony transmit path includes dynamics processing, 10 bands of parametric equalization, up to 1000 milliseconds of delay, a fader with gain control from +20 to -100 dB, and a line echo canceller. There is also a tone generator that is used to create DTMF digits and other call progress tones that may be sent to the telephone line and also played into the local room.

SoundStructure SR-Series Telco Processing

Telco Processing
Line echo cancellation, 80-3300Hz, 32msec tail-time
Dynamics processing: gate, expander, compressor, limiter, peak limiter on telco transmit and receive
Up to 8th order highpass and lowpass filters
1st or 2nd order high shelf and low shelf filters
10-bands of parametric equalization on telco transmit and receive
Call progress detection
Signal fader gain: +20 to -100 dB
Automatic gain control: +15 to -15dB on telco receive
Signal delay on telco transmit and receive: up to 1000 msec
Noise cancellation: 0-20dB noise reduction on telco receive

On the telephony receive path, the processing includes up to 20 dB of noise cancellation, automatic gain control, dynamics processing, 10-band parametric equalization, fader, and audio delay. In addition there is a call progress detector that analyzes the telephony input signal and reports if any call progress tones are present. For example, if the telephony line is busy, the phone rings.

.SoundStructure SR-Series Telco Processing


Telco Processing
1
2
N
Input Processing
Input Processing
Input Processing
Matrix
Output Processing
Output Processing
Output Processing
1
2
N
Submix Processsing
Telephony Processing
To Telco from Matrix
Dynamnics Processing
Parametric Equalization
Delay
Fader
Tone Generator
Line Echo Cancellation
D/A Converter
Analog Gain
Output to PSTN Line
From Telco to Matrix
Fader
Parametric Equalization
Dynamics Processing
Automatic Gain Control
Noise Cancellation
A/D Converter
Analog Gain
Input from PSTN Line
Call Progress Detection

Typically, the telephony cards are used in the C-series devices for audio conferencing applications. The telephony cards are also supported on the SR-series allowing additional plug-in cards for multiple audio conferencing telephone lines when C-series products are used with SR-series products. In some commercial sound applications it is also useful to have telephony access to either broadcast or monitor the audio in the system. Audio conferencing applications do not work with only SR-series devices because there is no acoustic echo cancellation processing in the SR-series devices.

Note: Using Telephony Cards with the SR-Series

The telephony cards should not be used with the SR-series of products for audio conferencing applications (i.e., simultaneous two-way audio communication) unless all the microphones in the system are connected to SoundStructure C-series devices. The SR-series products do not have acoustic echo cancellation.

Introducing SoundStructure Design Concepts

Before creating designs for the SoundStructure devices, the concepts of physical channels, virtual channels, and virtual channel groups are introduced. These concepts form the foundation of SoundStructure audio designs. In addition, the concepts of defining control virtual channels and control array virtual channels from the logic input and output pins are introduced.

Understanding Device Inputs and Outputs

All audio devices have inputs and outputs that are used to connect to other devices such as microphones and audio amplifiers. These inputs and outputs are labeled on the front or rear-panel (depending on the product) with specific channel numbers, such as inputs 1, 2, 3, etc., and these labels refer to particular inputs or outputs on the device. For instance, it is common to connect to input “1” or output “3” of an audio device. This naming convention works well -- meaning that it provides a unique identifier, or name, for each input and output -- as long as only a single device is used. As soon as a second device is added, input “1” no longer uniquely identifies an input since there are now two input 1’s if a system is made from two devices.

Traditionally, to uniquely identify which input “1” is meant, there’s additional information required such as a device identification name or number, requiring the user to specify input “1” on device 1 or input “1” on device 2 in order to uniquely identify that particular input or output. This device identification is also required when sending commands to a collection of devices to ensure the command affects the proper input or output signal on the desired device.

As an example, consider what must happen when a control system is asked to mute input 1 on device 1. The control system code needs to know how to access that particular input on that particular device. To accommodate this approach, most audio systems have an API command structure that requires specifying the particular device, perhaps even a device type if there are multiple types of devices being used, and, of course, the particular channel numbers to be affected by the command. This approach requires that the designer manually configure the device identification for each device that is used and take extra care to ensure that commands are referencing that exact input or output signal. If device identification numbers are changed or different inputs or outputs are used from one design to the next, this requires changing the control system code programming and spending additional time debugging and testing the new code to ensure the new device identifications and channel numbers are used properly. Every change is costly and is error prone, and can often delay the completion of the installation.

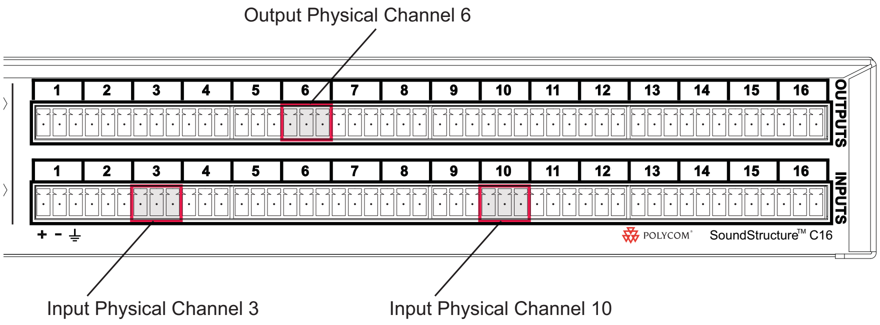
SoundStructure products have taken a different, and simpler, approach to labeling the inputs and outputs when multiple devices are used together. SoundStructure products achieve this simplification through the use of physical channels, virtual channels, and OBAM’s intelligent linking scheme. As shown in the Understanding Physical Channels section, physical channels are the actual input and outputs numbers for a single device and this numbering is extended sequentially when multiple devices are used. Understanding Virtual Channels extends this concept by creating a layer over physical channels that allows the physical channels to be referenced by a user defined label, such as “Podium mic”, rather than as a channel number.

Understanding Physical Channels

SoundStructure defines physical channels as a channel that corresponds to the actual inputs or outputs of the SoundStructure system. Physical channels include the SoundStructure analog inputs, analog outputs, submixes, the telephony interfaces, the conference link channels, and the logic input and output pins.

An example of physical channels is input 3, which corresponds to the physical analog input 3 on the rear-panel of a SoundStructure device, input 10 (corresponds to analog input 10), and output 6, which corresponds to the physical analog output 6 on a SoundStructure device, as shown in the following figure.

Example of Physical Input Channels


Output Physical Channel 6
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
OUTPUTS
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
INPUTS
+ - ⏚
POLYCOM
SoundStructure™ C16
Input Physical Channel 3
Input Physical Channel 10

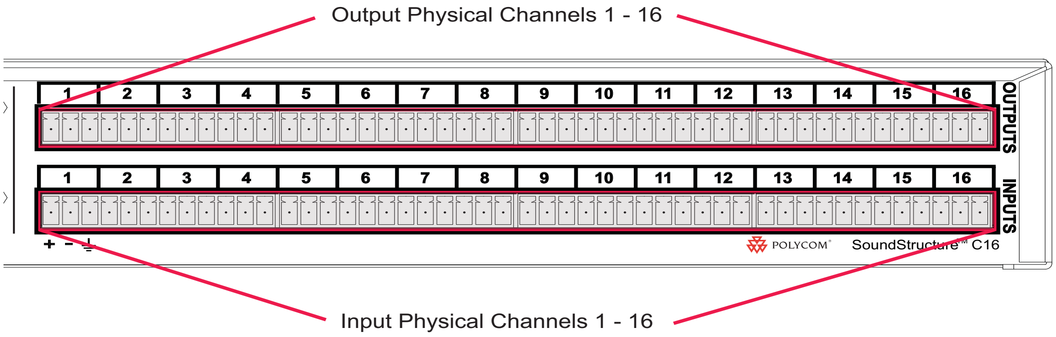
When designing with SoundStructure products, the analog inputs (such as microphones, or other audio sources) and outputs from the system (such as audio sent to amplifiers) connect to SoundStructure’s physical channels.

The physical input channels and the physical output channels are numbered from 1 to the maximum number of physical channels in a system. As described below, this approach is an enhancement of how traditional audio signals are labeled and how their signals are uniquely referenced.

Numbering Physical Channel On A Single SoundStructure Device

As described previously, in single-device SoundStructure installations (for example using a single SoundStructure C16), the physical channel numbering for the inputs and outputs corresponds to the numbering on the rear-panel of the device. For example, as shown in the following figure, physical input channel 3 corresponds to input 3 on the SoundStructure C16 device.

Example of Corresponding Physical Channels on a Single SoundStructure Device



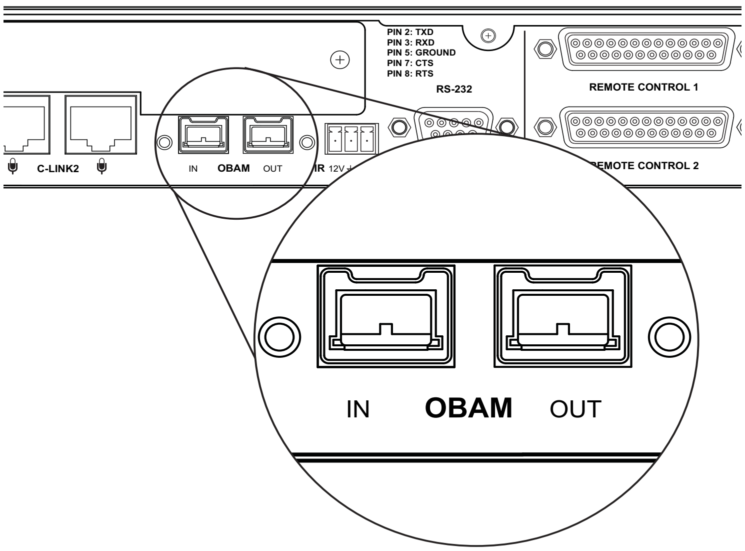
Numbering Physical Channel With Multiple SoundStructure Devices

When multiple SoundStructure devices are linked using One Big Audio Matrix (OBAM) to form a multi-device SoundStructure system, instead of using a device identification number, the physical channel numbering for both the inputs and the outputs ranges from 1 to the maximum number of inputs and outputs, respectively, in the system. This is an extension of the single device setup where the physical channel numbers for channels on the second device are the next numbers in the sequence of inputs from the first device. For if there are two devices and the first device is a SoundStructure C16, the first input on the second device becomes physical input 17. This continuation of the sequence of numbers is possible due to the design of the OBAM Link interface.

OBAM Link is the method for connecting multiple devices together by connecting the OBAM Link cable from one device to the next. The following figure shows the location of the OBAM connections and the OBAM OUT and OBAM IN connections on the rear-panel of a SoundStructure device. To help verify when the OBAM Link is connected properly, there are status LEDs near the outer edge of each connector that illuminate when the devices are linked successfully.

The OBAM link is bidirectional - data flows in both an upstream and downstream direction meaning that the bus does not need to be looped back to the first device.

OBAM Connections on a SoundStructure Device

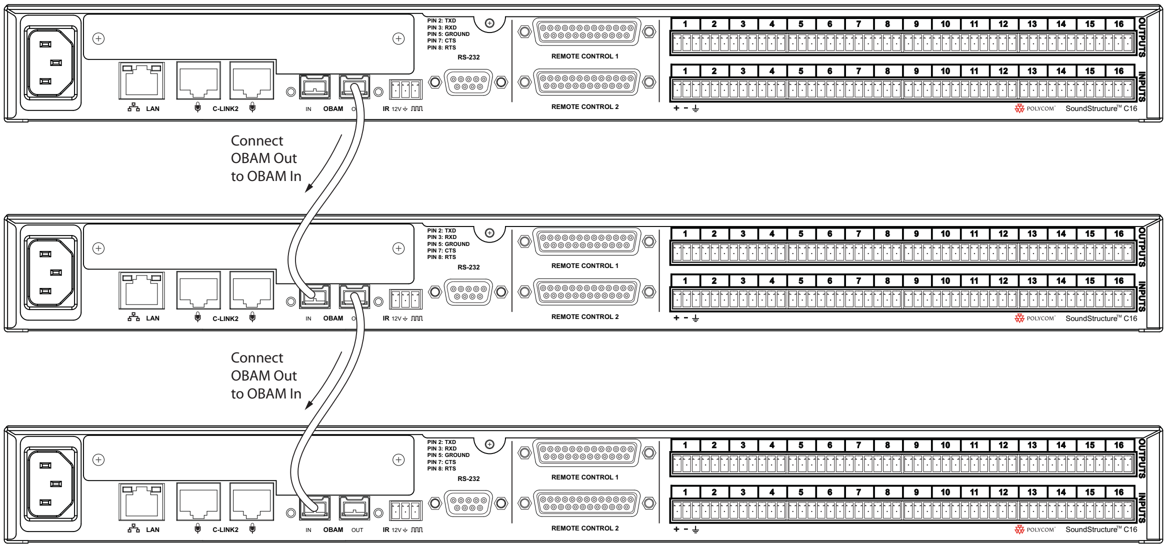


When multiple devices are linked together via OBAM, the SoundStructure devices communicate to each other, determine which devices are linked and automatically generate internal device identifications. These device identifications are sequential from the first device at device ID 1 through the latest device linked over OBAM. Externally, there are no SoundStructure device identifications that must to be set or remembered. The internal device identifications are not required by the user/designer and are not user settable.

As described previously, rather than referring to physical channels on different devices by using a device identification number and a local physical input and output number, SoundStructure devices are designed so that the physical channel numbering is sequential across multiple devices. This allows one to refer to different channels on multiple devices solely by using a physical channel number that ranges from 1 to the maximum number of channels in the linked system. As shown next, how the devices are OBAM linked determines the resulting numbering of the physical channels for the overall system.

To properly link multiple SoundStructure devices, connect the OBAM OUT port on the first device (typically the top SoundStructure device in the equipment rack) to the OBAM IN port on the next SoundStructure device and continue for additional devices. This connection strategy, shown in the following figures, simplifies the sequential physical channel numbering as described next.

OBAM Connection Strategy for SoundStructure Devices



Once multiple devices are OBAM linked, it is easy to determine the system's input and output physical channel numbering based on the individual device’s physical channel numbering. The way the physical channels in a multiple device installation are numbered is as follows:

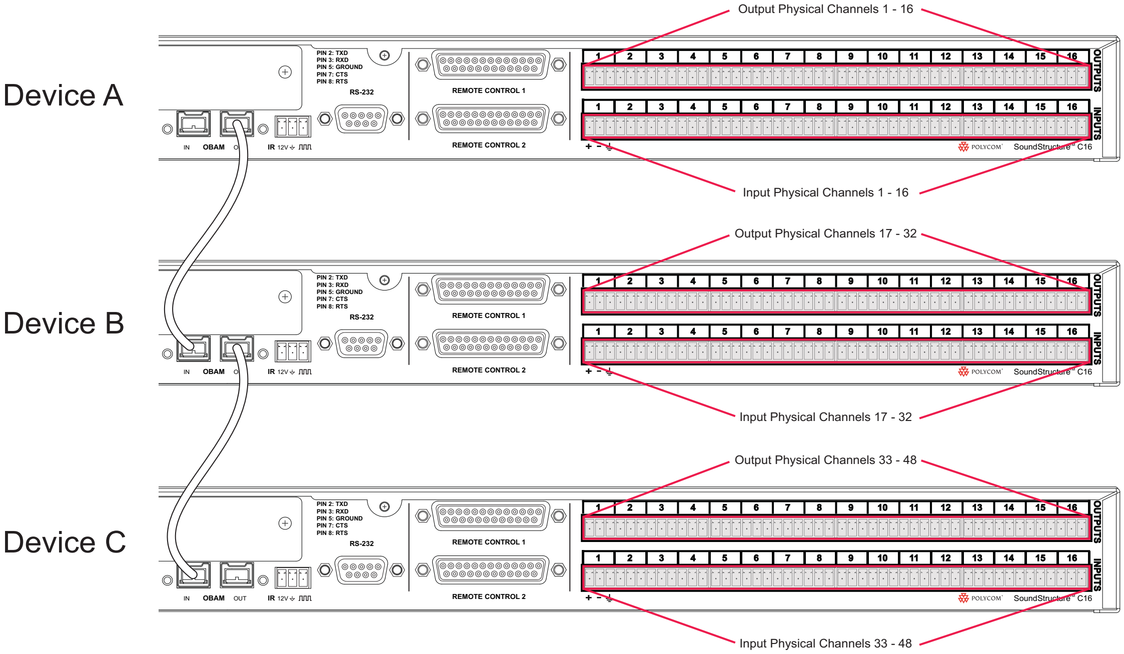
1. The SoundStructure device that only has a connection on the OBAM OUT connection (recommended to be the highest unit in the rack elevation) is the first device and its inputs and outputs are numbered 1 through N where N is the number of inputs and outputs on the device (for instance, 16 inputs for a SoundStructure C16 device).
2. The SoundStructure device whose OBAM IN port is connected to the OBAM OUT connection of the previous device becomes the next M inputs and outputs for the system where M is the number of inputs and outputs on the second device (for instance, 12 inputs for a SoundStructure C12 device).
3. This continues until the last device in the link which has an OBAM IN connection to the unit above it and has no connection on the OBAM OUT port.

Note: OBAM Linking Devices

It is recommended that the units be linked together in the top-down order connecting the higher OBAM OUT connection to the next OBAM IN connection. One way to remember this ordering is to imagine the data flowing downhill out of the top unit and into the next unit and so on.

Following the connections in the previous figure, consider the system of three SoundStructure C16 devices shown in the following figure as an example of this linking order and how the physical channels are numbered. In this example the OBAM output of device A is connected to the OBAM input of device B and the OBAM output of device B is connected to the OBAM input of device C. While the individual devices have physical channel inputs ranging from 1 to 16 and physical outputs ranging from 1 to 16, when linked together, the physical inputs and outputs of the overall system are numbered 1 to 48. These physical channel numbers of all the inputs and outputs are important because the physical channel numbers are used to create virtual channels, as discussed in the next section.

Physical Channels Numbering when OBAM Linked


Output Physical Channels 1 - 16
Device A
Input Physical Channels 1 - 16
Output Physical Channels 17 - 32
Device B
Input Physical Channels 17 - 32
Output Physical Channels 33 - 48
Device C
Input Physical Channels 33 - 48
OUTPUTS
INPUTS
RS-232
REMOTE CONTROL 1
REMOTE CONTROL 2
IN
OBAM
OUT
IR
SoundStructure C16

With the linking of devices as shown in the previous figure, the physical channels are ordered as expected and shown in that figure and summarized in the following table.

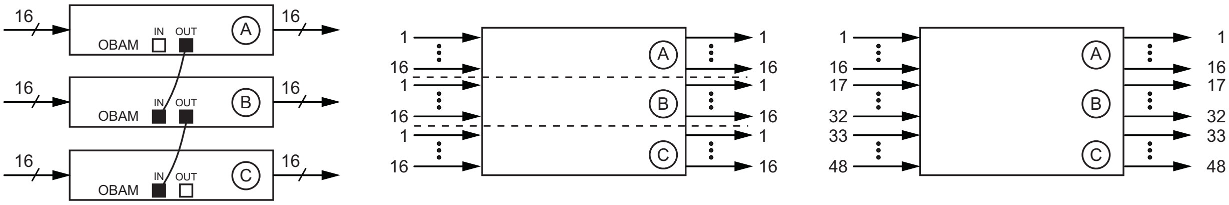
Device A's inputs and outputs become the first sixteen physical inputs and sixteen outputs on the system, device B's inputs and outputs become the next sixteen physical inputs and next sixteen physical outputs on the system, and device C's inputs and output become the last sixteen physical inputs and sixteen physical outputs on the system.

Local and System Input and Output Numbering for OBAM Linked SoundStructure Devices

Device	*Local Numbering (input and output)*	*System Numbering (input and output)*
A	1 - 16	1 - 16
B	1 - 16	17 - 32
C	1 - 16	33 - 48

The system built from the top-to-bottom, OBAM out-to-OBAM-in linking results in a simple way of numbering the physical input and output connections in a simple linear sequential fashion. Conceptually, the linking of these devices should be viewed as creating one large system from the individual systems, as shown in the next figure.

Viewing OBAM Linked Devices as One Large System

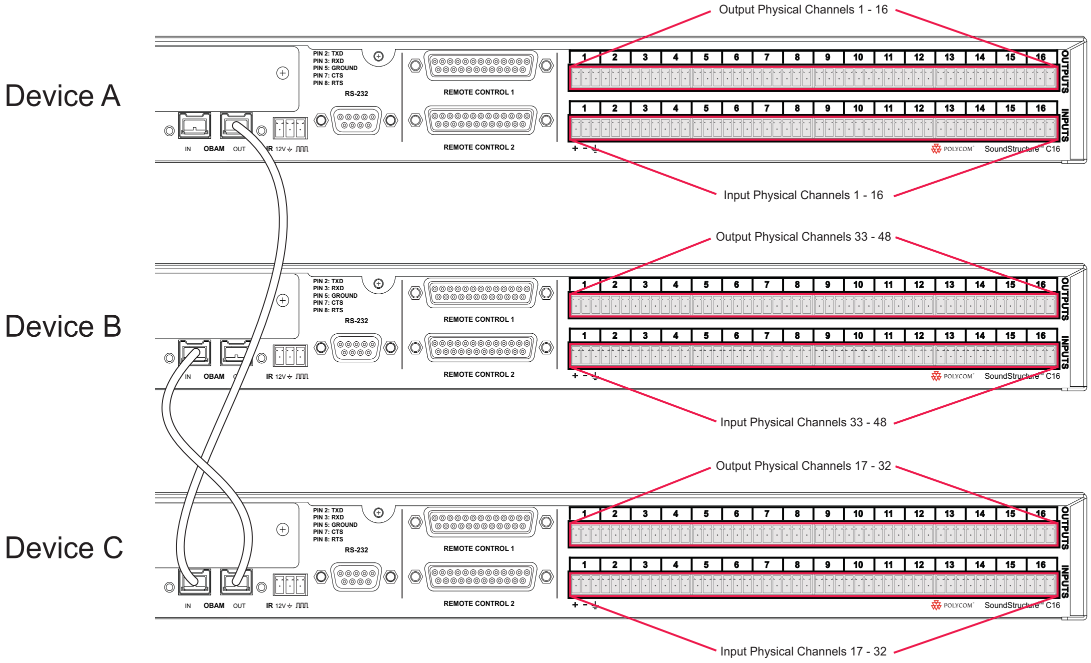

16
16
OBAM
IN
OUT
A
16
16
OBAM
IN
OUT
B
16
16
OBAM
IN
OUT
C
1
16
1
16
1
16
A
B
C
1
16
1
16
1
16
1
16
17
32
33
48
A
B
C
1
16
17
32
33
48

Note: Numbering Physical Channels in a Multi-Device System

The numbering of the physical channels in a multi-device system is determined by how the devices are linked over OBAM. Changing the OBAM link cabling after a system has been designed and uploaded to the devices causing the system to not operate properly.

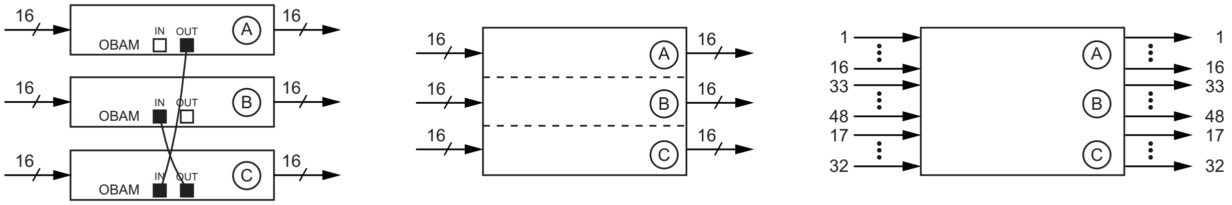
If multiple devices are OBAM linked in a different order, the numbering of the physical channels is different. As an example of what not to do, consider the following figure where device C is connected to both device A and to device B. Based on the physical ordering algorithm described previously, device A only has an OBAM OUT connection which makes this device the first device in the link. Next, device C becomes the second device in the link and finally device B becomes the third device in the link. The result is that the inputs and outputs on device C become inputs 17-32 and outputs 17-32 on the full system even though device B is physically installed on top of device C.

Example of SoundStructure Devices OBAM Linked Out of Order


Output Physical Channels 1 - 16
Device A
Input Physical Channels 1 - 16
Output Physical Channels 33 - 48
Device B
Input Physical Channels 33 - 48
Output Physical Channels 17 - 32
Device C
Input Physical Channels 17 - 32
OBAM
IN
OUT
IR
RS-232
REMOTE CONTROL 1
REMOTE CONTROL 2
OUTPUTS
INPUTS
POLYCOM
SoundStructure C16

Conceptually, this creates a system similar to the system as shown in the next figure and summarized in the following table.

Example of SoundStructure Devices OBAM Linked Out of Order



The organization of the devices in this example would make it confusing to properly terminate inputs and outputs to the desired physical inputs and outputs. Any OBAM linking scheme other than the out-to-in, top-to-bottom system, is not recommended as it can increase system debug and installation time.

Local and System Numbering for SoundStructure Devices OBAM Linked Out of Order

Device	Local Numbering	System Numbering
A	1 - 16	1 - 16
B	1 - 16	33 - 48
C	1 - 16	17 - 32

Due to this possible confusion of the numbering of physical inputs and outputs, always connect the devices as recommended in the top-down order connecting the higher OBAM OUT connection to the next OBAM IN connection.

Physical Channel Summary

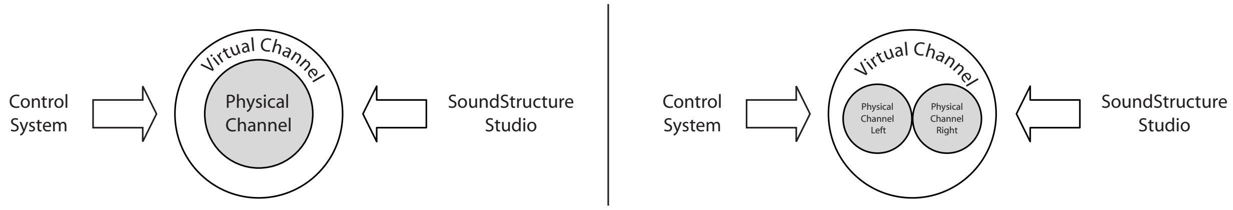
Physical channels and the OBAM Link were introduced in the previous section as a simplification of how to refer to the actual physical inputs and outputs when multiple SoundStructure devices are used. By OBAM Linking multiple SoundStructure devices in an OBAM out-to-OBAM-in fashion from top to bottom, the physical channel numbers in a multi-unit installation are sequential from 1 to the maximum number of inputs and outputs in the system. No longer is a specific device identification required to uniquely identify which input “1” is meant when there are multiple devices. When multiple SoundStructure devices are used, there is only one input “1” and it corresponds to the first input on the top device. The first input on the second device is input 17 (if the first device is a SoundStructure C16).

In the next section, the concept of physical channels is extended as the new concept of *virtual channels* is introduced as a way to easily and more flexibly reference the physical input and output channels, simplifying both SoundStructure device setup and how SoundStructure devices are controlled with external control systems.

Understanding Virtual Channels

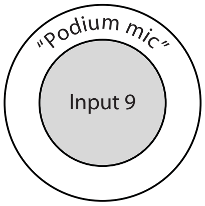
A virtual channel can be thought of as a layer that is wrapped around one or more physical channels. A virtual channel can represent either an individual physical channel or it can represent a collection of strongly associated physical channels, such as a stereo pair of signals as shown in the following figure.

SoundStructure Studio Virtual Channels



Virtual channels are created by specifying a virtual channel name, one or more physical channels, and a type of virtual channel. Once defined, the virtual channel name becomes the primary way of referring to that particular input or output instead of using the physical channel number. For example, an A/V designer defines the virtual channel that is connected to input physical channel 9 as “Podium mic,” as shown in the following figure. From then on, any settings that need adjusting on that input are adjusted by controlling the virtual channel “Podium mic”. The association between the virtual channel and the underlying physical channel or channels means that you can think of virtual channels as describing how the system is wired.

Virtual Channel Naming



Note: Naming Virtual Channels

The virtual channel name is case-sensitive and needs to have the quotes around the text. “Podium mic”, “Podium Mic”, and “PODIUM mic” would represent different virtual channels.

The main benefit of virtual channels is that once a SoundStructure design is created and the virtual channels have been defined, it is possible to change the particular physical input or output used by moving the physical connection on the rear-panel of the SoundStructure device and redefining the virtual channel to use the new physical input or output that is used. Because any control system code must use the virtual channel name, the control source code does not have to change even if the actual wiring of the physical inputs or outputs change. By using virtual channel names the controller code controls (for example, mutes or changes volume) the SoundStructure devices through the virtual channel names, not the underlying physical input and output that a particular audio signal is connected to.

For instance, if a virtual channel were named “Podium mic” then the control system code would control this channel by sending commands to “Podium mic”. It would not matter to the control system if on one installation “Podium mic” were wired to input 1 and on another installation “Podium mic” was wired to input 17. The same control system code can be used on both installations because the SoundStructure devices translate the virtual channel reference to the underlying physical channel(s) that were specified when the virtual channel was defined. By using the same API commands on different systems that refers to “Podium mic”, the control system code is insulated from the actual physical connections which are likely to change from one installation to the next. The virtual channel definition makes the design portable and easily reusable.

The use of virtual channels also improves the quality of the control system code because it is easier to write the correct code the first time as it is more difficult to confuse “Podium mic” vs. “VCR audio” in the code than it would be to confuse input 7 on device 2 vs. input 9 on device 1. The clarity and transparency of the virtual channel names reduces the amount of debugging and subsequently the amount of time to provide a fully functional solution.

Another benefit of working with virtual channels is that stereo signals can be more easily used and configured in the system without having to manually configure both the left and right channels independently. As shown later in the guide, the SoundStructure Studio software automatically creates the appropriate monaural mixes when interfacing a stereo signal to mono destination and vice versa.

Using virtual channels that represent stereo physical signals reduces the chance of improper signal routings and processing selections. The net result is that both designs and installations can happen faster and with higher quality.The motivation for using virtual channels is to make the system reusable across different installations regardless of how the system is wired because the SoundStructure device knows how to translate commands that are sent to virtual channels, such as “Podium mic”, to the appropriate underlying physical channel.

Note: Defining Virtual Channels

Virtual channels are a high-level representation that encompasses information about the physical channel. Virtual channels are used to configure and control the underlying physical channel(s) without having to know the underlying physical channel numbers.

Virtual Channel Summary

Virtual channels are a new concept introduced for SoundStructure products that makes it possible to refer to one or more physical channels at a higher level by creating a virtual channel and a memorable virtual channel name.

Using SoundStructure virtual channels is the only way to configure and control the underlying physical channels with third-party control systems. The physical input and output channel numbering described in the section Understanding Physical Channels is used only in the definition of virtual channels so that the virtual channel knows which physical channel(s) it refers to.

By using virtual channel names rather than hard wiring physical input and output channels in the control system code, the control system source code is more portable across other installations that use the same virtual channel names regardless of which physical channels were used to define the virtual channels (in other words, how the system is wired).

Virtual channels also simplify the setup and configuration of a system because it is easier to understand and view changes to Podium mic than it is to have to refer to a signal by a particular physical input or output number such as input 17.

Virtual channels are defined by SoundStructure Studio during the project design steps using the vcdef command described in Appendix A. As an example, a mono virtual channel that is connected to physical input 8 would be defined as:

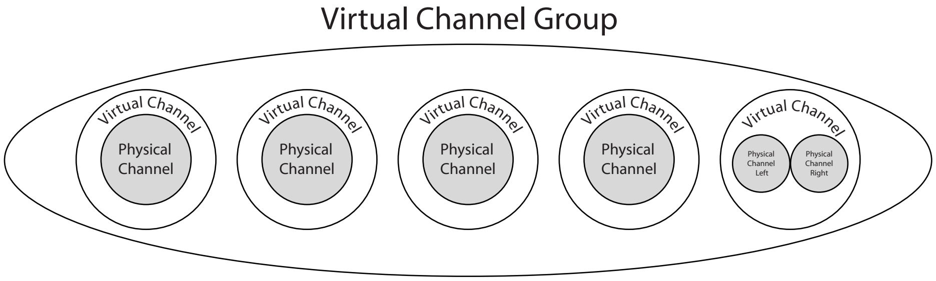
```
vcdef “Podium mic” mono cr_mic_in 8
```

Understanding Virtual Channel Groups

It is often convenient to be able to refer to a group of virtual channels and control a group of virtual channels with a single command. Virtual channel groups are used with SoundStructure products to create a single object made up of loosely associated virtual channels. Once a virtual channel group has been created, all commands to a virtual channel group affect the virtual channels that are part of the virtual channel group and command acknowledgments from all the members of the virtual channel group returned. In addition the virtual channel group returns an acknowledgment that is the value of the acknowledgment of the first member of the group.

Virtual channel groups are a wrapper around a number of virtual channels, as shown in the following figure.

A Virtual Channel Group


Virtual Channel Group
Virtual Channel
Physical Channel
Virtual Channel
Physical Channel
Virtual Channel
Physical Channel
Virtual Channel
Physical Channel
Virtual Channel
Physical Channel Left
Physical Channel Right

As an example of a virtual channel group, consider in the next figure the creation of the virtual channel group “Mics” made up of the entire collection of individual microphone virtual channels in a room. Once the virtual channel group “Mics” has been created, it is possible to configure and control all the microphones at the same time by operating on the “Mics” virtual channel group.

If the group “Mics” is muted with the command:

```
set mute “Mics” 1
```

then the acknowledgments returned from the SoundStructure device are:

```
val mute “Wireless mic” 1
val mute “Table mic 1” 1
val mute “Table mic 2” 1
val mute “Table mic 3” 1
val mute “Table mic 4” 1
val mute “Table mic 5” 1
val mute “Table mic 6” 1
val mute “Table mic 7” 1
val mute “Table mic 8” 1
val mute “Podium mic” 1
val mute “Mics” 1
```

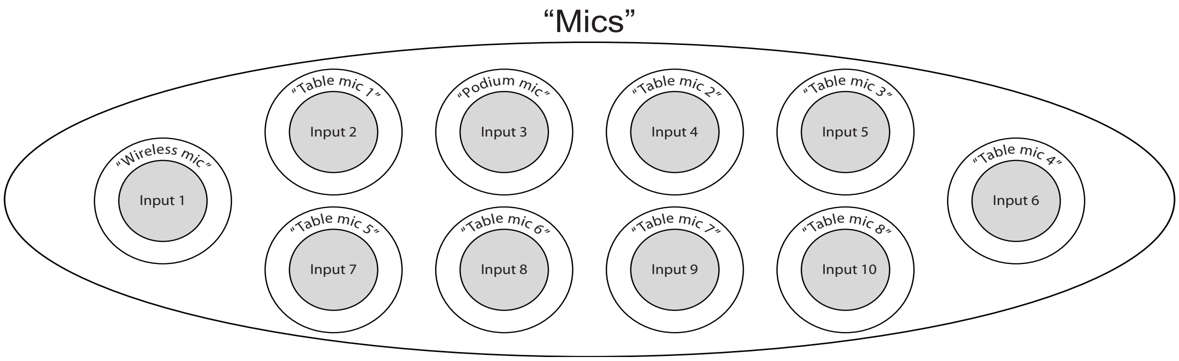
The final command acknowledgment value for the group “Mics” is the value returned from the first member of the virtual channel group “Mics”.

It is possible to have multiple virtual channel groups that include the same virtual channels. Commands sent to the particular virtual channel group affect the members of the group and all members of the group respond with the appropriate command acknowledgments.

Note: Virtual Channels Include in Multiple Groups

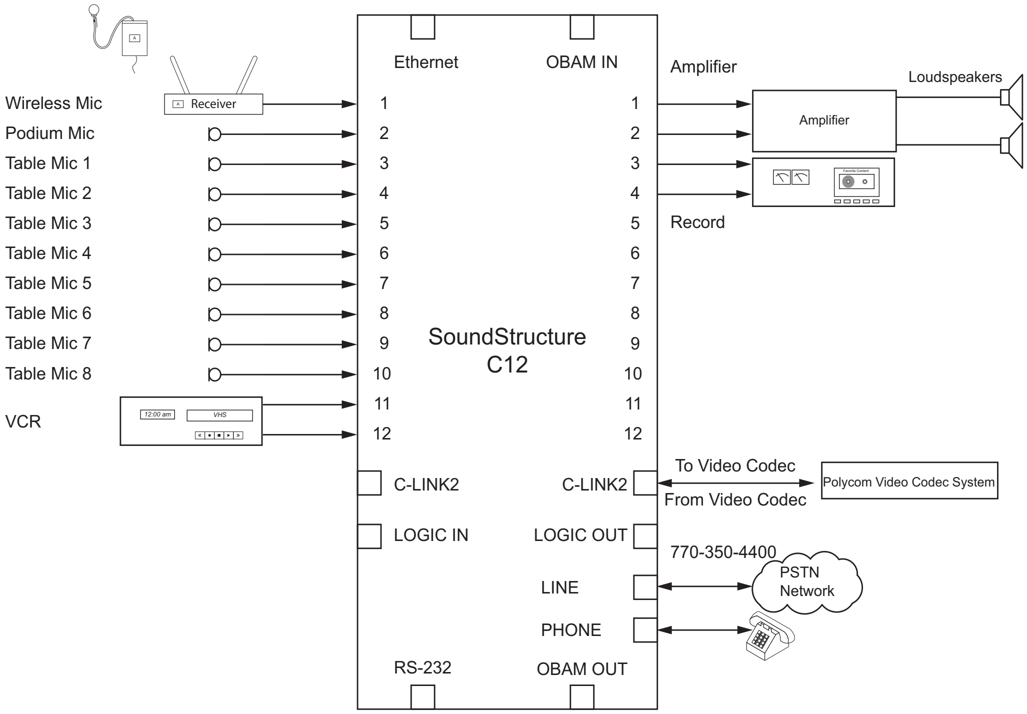
Multiple virtual channel groups may include the same virtual channels, in other words, a virtual channel can belong to more than one virtual channel group.

A Virtual Channel Group


“Mics”
“Wireless mic”
Input 1
“Table mic 1”
Input 2
“Podium mic”
Input 3
“Table mic 2”
Input 4
“Table mic 3”
Input 5
“Table mic 4”
Input 6
“Table mic 5”
Input 7
“Table mic 6”
Input 8
“Table mic 7”
Input 9
“Table mic 8”
Input 10

As an example of using physical channels, virtual channels, and virtual channel groups, consider a SoundStructure C12 device where there are ten microphone inputs, a telephony interface, and a Polycom Video Codec system as shown in the following figure.

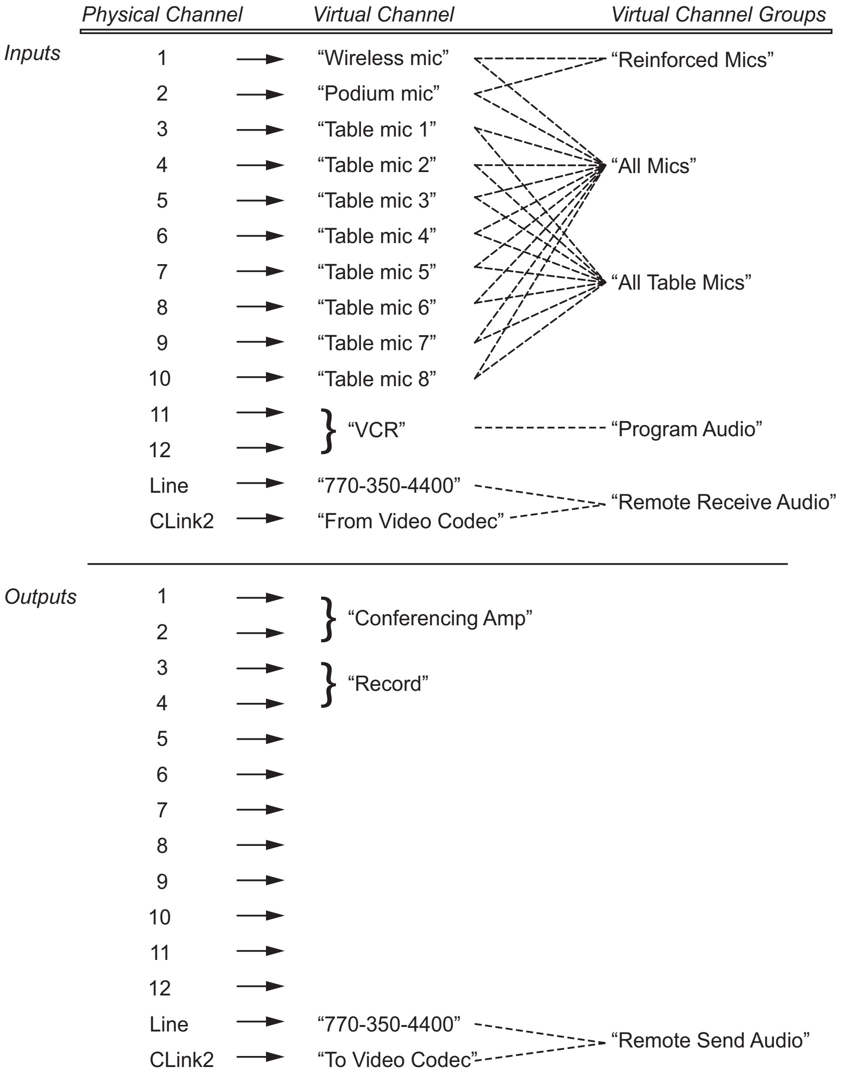
SoundStructure C12 with Physical Channels, Virtual Channels, and Virtual Channel Groups


Ethernet
OBAM IN
Amplifier
Loudspeakers
Wireless Mic
Receiver
Podium Mic
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
VCR
12:00 am
VHS
1
2
3
4
5
6
7
8
9
10
11
12
SoundStructure
C12
1
2
3
4
5
6
7
8
9
10
11
12
Amplifier
Record
C-LINK2
C-LINK2
To Video Codec
From Video Codec
Polycom Video Codec System
LOGIC IN
LOGIC OUT
770-350-4400
PSTN
Network
LINE
PHONE
RS-232
OBAM OUT

In the above example, there is a wireless microphone and a podium microphone, both reinforced into the room, eight table top microphones, and a stereo VCR for audio playback. As shown in this figure the system is wired with the wireless microphone in input 1, the podium mic on input 2, the table mics 1-8 on inputs 3-10, a stereo VCR is connected to inputs 11 and 12 and a Polycom Video Codec is connected over the digital ConferenceLink interface.

Virtual channel definitions are defined, as shown in the following figure.

Virtual Channel Definitions


Physical Channel
Virtual Channel
Virtual Channel Groups
Inputs
1 "Wireless mic"
2 "Podium mic"
3 "Table mic 1"
4 "Table mic 2"
5 "Table mic 3"
6 "Table mic 4"
7 "Table mic 5"
8 "Table mic 6"
9 "Table mic 7"
10 "Table mic 8"
11, 12 "VCR"
Line "770-350-4400"
CLink2 "From Video Codec"
"Reinforced Mics"
"All Mics"
"All Table Mics"
"Program Audio"
"Remote Receive Audio"
Outputs
1, 2 "Conferencing Amp"
3, 4 "Record"
5
6
7
8
9
10
11
12
Line "770-350-4400"
CLink2 "To Video Codec"
"Remote Send Audio"

The virtual channel definitions make it easy to work with the different signals since each virtual channel has a specific name and refers to a particular input or output. For instance to take the phone off hook, commands are sent to the "770-350-4400" virtual channel in this example. If there were multiple telephony interfaces, each telephony interface would have its own unique virtual channel definition. It is possible to create a virtual channel group of multiple telephony virtual channels so all systems could be put onhook together at the end of a call, etc.

In this example there are several virtual channel groups defined including "Reinforced Mics", "All Mics", "All Table Mics", "Program Audio", "Remote Receive Audio", and "Remote Send Audio".

Virtual Channel Group Summary

Virtual channel groups are an easy way to create groups of signals that may be controlled together by sending an API command to the virtual channel group name. It is possible to have more than one virtual channel group and to have the same virtual channel in multiple virtual channel groups. It is also easy to add or remove signals from the virtual channel group making virtual channel groups the preferred way of controlling or configuring multiple virtual channels simultaneously.

Virtual channel groups re defined by SoundStructure Studio during the project design steps using the `vcgdef` command described in Appendix A. As an example, a virtual channel group with two members, Table Mic 1 and Table Mic 2, would be defined as:

```
vcgdef “Zone 1” “Table Mic 1” “Table Mic 2”
```

Understanding Telephone Virtual Channels

Telephony virtual channels are created with the telephony inputs and telephony outputs - each direction on a telephony channel is used to create a virtual channel. There are two types of physical channels used: `pstn_in`, and `pstn_out`, in the definition of telephony virtual channels.

By default, SoundStructure Studio creates virtual channel definitions for both the input and output commands. The command set in Appendix A shows which commands operate on the telephone output virtual channels and which operate on the telephony input channels.

For example, the `phone_connect` and `phone_dial` commands operate on the telephony output channel while the `phone_dial_tone_gain` command operates on the telephone input channel.

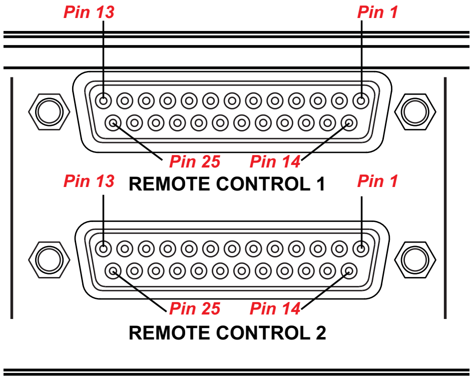
Defining Logic Pins

SoundStructure logic input and output pins are also considered physical inputs and outputs that can be abstracted with control virtual channels and control array virtual channels.

Labeling Physical Logic Pins

The physical logic pins and labeling are shown in the following figure.

Physical Logic Pins and Labeling



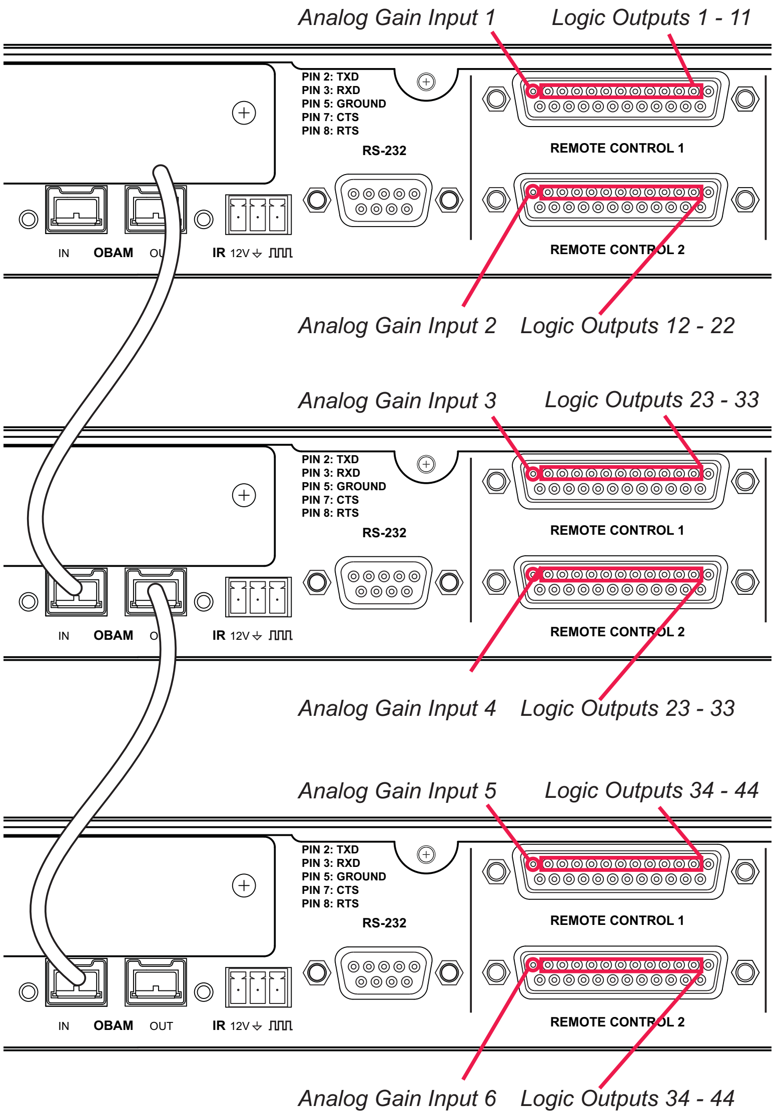
SoundStructure Logic

Pin	Signal	Pin	Signal
REMOTE CONTROL 1			
1	+5V	14	Logic input 1
2	Logic output 1	15	Logic input 2
3	Logic output 2	16	Logic input 3
4	Logic output 3	17	Logic input 4
5	Logic output 4	18	Logic input 5
6	Logic output 5	19	Logic input 6
7	Logic output 6	20	Logic input 7
8	Logic output 7	21	Logic input 8
9	Logic output 8	22	Logic input 9
10	Logic output 9	23	Logic input 10
11	Logic output 10	24	Logic input 11
12	Logic output 11	25	Ground
13	Analog gain input 1		
REMOTE CONTROL 2			
1	+5V	14	Logic input 12
2	Logic output 12	15	Logic input 13
3	Logic output 13	16	Logic input 14
4	Logic output 14	17	Logic input 15
5	Logic output 15	18	Logic input 16
6	Logic output 16	19	Logic input 17
7	Logic output 17	20	Logic input 18
8	Logic output 18	21	Logic input 19
9	Logic output 19	22	Logic input 20
10	Logic output 20	23	Logic input 21
11	Logic output 21	24	Logic input 22
12	Logic output 22	25	Ground
13	Analog gain input 2		

The logic inputs and logic outputs have physical inputs and outputs 1 - 11 on Remote Control 1 connector and 12 - 22 on Remote Control 2 connector on each SoundStructure device.

When multiple devices are OBAM linked, as shown in the next figure, the logic inputs and outputs on the first device are numbered 1 - 22 and the logic inputs and outputs on the second device (device B) are numbered 23 - 44, and so on. The analog gain inputs are numbered 1 and 2 on the first device, 3 and 4 on the second device, and so on.

Numbering of Logic Inputs and Outputs on an OBAM Linked SoundStructure Device

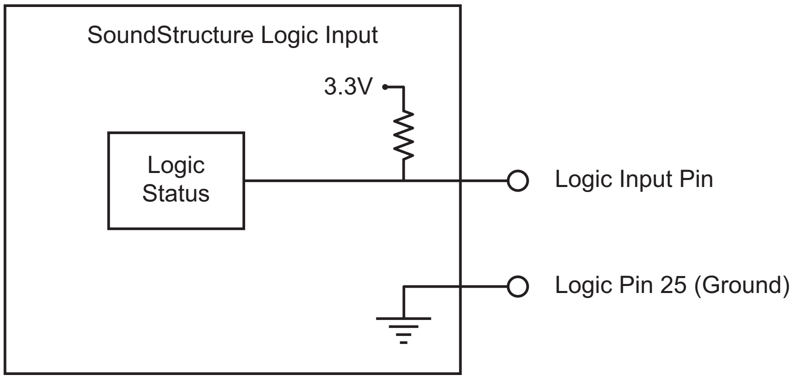

Analog Gain Input 1
Logic Outputs 1 - 11
PIN 2: TXD
PIN 3: RXD
PIN 5: GROUND
PIN 7: CTS
PIN 8: RTS
RS-232
REMOTE CONTROL 1
REMOTE CONTROL 2
IN
OBAM
OUT
IR 12V
Analog Gain Input 2
Logic Outputs 12 - 22
Analog Gain Input 3
Logic Outputs 23 - 33
Analog Gain Input 4
Logic Outputs 23 - 33
Analog Gain Input 5
Logic Outputs 34 - 44
Analog Gain Input 6
Logic Outputs 34 - 44

Due to the one large system design philosophy, logic input pins on any device can be used to control features on any SoundStructure device - not just provide control on the device the logic inputs are on. Similarly logic outputs can be used to provide status on signals on any SoundStructure device - not just status on a physical channel on that particular device.

Logic Inputs

All digital logic inputs (logic inputs 1 - 22) operate as contact closures and can be connected to ground (closed) or not connected to ground (open). The logic input circuitry is shown in the following figure. The default value for logic inputs is 1 due to the pull up resistor. The value for the pin changes to 0 when the pin is shorted to ground. The value of the logic pin is read or written with the `digital_gpio_value` parameter. See Using Events, Logic, and IR and Appendix A: Command Protocol Reference Guide for more details.

Logic Input Circuitry for SoundStructure Devices

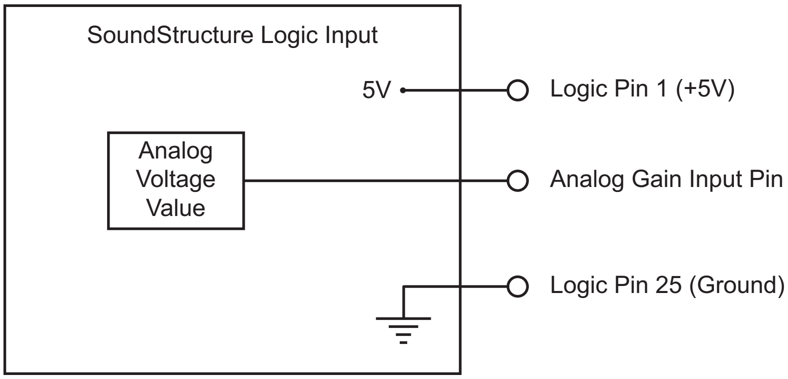


Analog Gain Input

The analog gain inputs (analog gain 1 and 2) operate by measuring an analog voltage between the analog input pin and the ground pin. The maximum input voltage level should not exceed +6 V. It is recommended that the +5 V supply on Pin 1 be used as the upper voltage limit.

The next figure shows the analog gain input pin and the associated +5 V and ground pins that are used with the analog gain input pin. The analog voltage on the analog gain input pin is converted to a digital value via an analog-to-digital converter for use within the SoundStructure devices. The maximum voltage value, that is, 0 dBFS on the analog gain input, is 4.096 V. 0V is converted to 0 and 4.096 V and above is converted to 255.

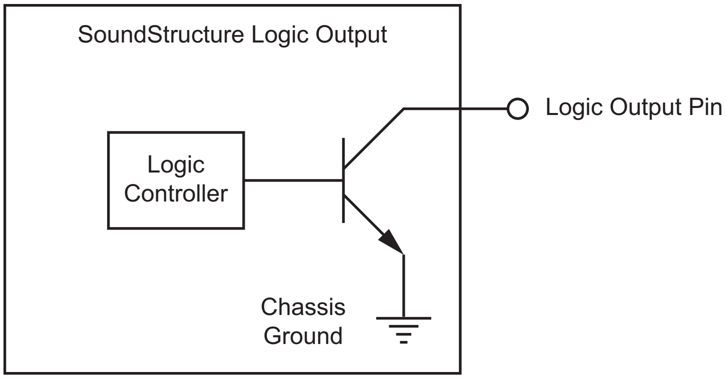
Analog Gain Input Pin for SoundStructure Devices


SoundStructure Logic Input
5V
Logic Pin 1 (+5V)
Analog Voltage Value
Analog Gain Input Pin
Logic Pin 25 (Ground)

Logic Outputs

All logic outputs are configured as open-collector circuits and can be used with external positive voltage sources. The maximum voltage that should be used with the logic outputs is 60 V. Each pin can sink up to 60mA. When using the internal 5V power supply, the maximum current that is supplied across all logic outputs on a SoundStructure device is 500 mA.

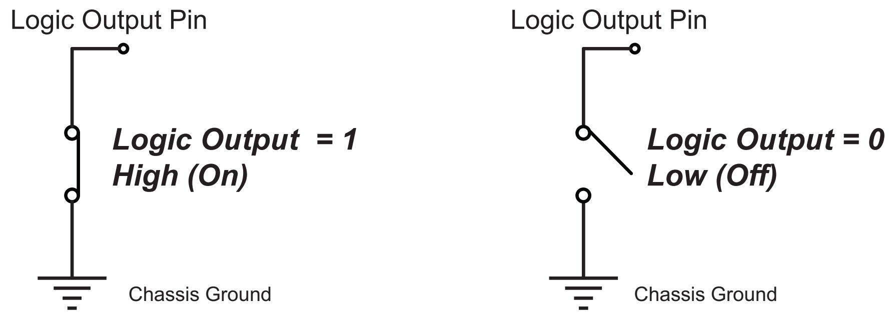
Logic Output Pin on SoundStructure Devices


SoundStructure Logic Output
Logic Output Pin
Logic Controller
Chassis Ground

The open collector design is shown in the following figure and works as a switch as follows: when the logic output pin is set **high** (on), the transistor turns on and the signal connected to the logic output pin is grounded and current flows from the logic output pin to chassis ground.

When the logic output is set **low** (off), the transistor turns off and an open circuit is created between the logic output and the chassis ground preventing any flow of current, as shown in the following figure.

Logic Output Pin Set to Low (Off)



Examples of using logic input and output pins may be found in Using Events, Logic, and IR of this guide.

Controlling Virtual Channels

The concept of virtual channels also applies to the logic inputs and outputs. The A/V designer can create control virtual channels that consist of a logic input or output pin.

Logic pins can be defined via the command line interface from SoundStructure Studio or a control terminal with the following syntax to define a logic input on logic input pin 1:

```
vcdef “Logic Input Example” control digital_gpio_in 1
```

which returns the acknowledgment

```
vcdef "Logic Input Example" control digital_gpio_in 1
```

A logic output pin definition using output pin 1 is created with the command:

```
vcdef "Logic Output Example" control digital_gpio_out 1
```

which returns the acknowledgment

```
vcdef "Logic Output Example" control digital_gpio_out 1
```

Once defined, the designer can refer to those control virtual channels by their name. As with the example above, the designer created a control input virtual channel “Logic Input Example”. The SoundStructure device can be queried with a control system to determine the value of the logic pin and when it is active, it could be used to change the status of the device. When the “Logic Input Example” input is inactive, it could, for example, be used with an external control system to unmute the microphones. In version 1.0 of the firmware logic pins must be queried by an external control system and then the control system can execute commands or a series of commands on the device.

The value of control virtual channels may be queried by the control system by using the command `digital_gpio_state`. An example of this is shown below.

```
get digital_gpio_state “Logic Input Example”
```

The state of digital logic output may also be set active using the d`igital_gpio_state` command as follows for the control virtual channel “Logic Output Example” that would be created with the `vcdef` command.

```
set digital_gpio_state “Logic Output Example” 1
```

Additional information about using logic pins may be found in Appendix A.

Controlling Array Virtual Channels

Multiple logic pins may be associated together with a control array virtual channel. Control array virtual channels are created by one or more logic input or logic output pins. Once a control array channel is defined, the value of the group of pins can be queried or set using the `digital_gpio_value` command.

The value of the digital control array is the binary sum of the individual logic pins. For example, if a control array virtual channel is defined with digital output pins 2, 3, and 4, then the value of the control array channel is in the range of 0 to 7 with physical logic pin 4 as the most significant bit and physical logic pin 2 as the least significant bit.

As an example, consider a control array named “logic array” that uses physical logic input pins 2, 3, and 4 that is created with the following syntax:

```
vcdef “logic array” control_array digital_gpio_in 2 3 4
```

which returns the command acknowledgment:

```
vcdef "logic array" control_array digital_gpio_in 2 3 4
```

In this example, three input pins have been specified with pin 2 first and pin 4 listed last. The value of the digital input array can be queried using the get action:

```
get digital_gpio_value "logic array"
val digital_gpio_value "logic array" 7
```

The value of the logic array depends on the value of the individual logic input pins 4, 3, and 2. A logic pin has a value of 0 when that the pin is shorted to ground and a value of 1 when that pin is open.

The order that the pins are listed in the control array definition is defined so that the last pin specified is the most significant bit and the first pin specified is the least significant bit. For the example above where the control array was defined with pins 2 3 4, the 3-bit value is formed by using pin 4 as the most significant bit, pin 3 as the next bit, and pin 2 as the least significant bit.

Control Array and Logic Input Pin Values

Control Array Value	Pin 4	Pin 3	Pin 2
7	1	1	1
6	1	1	0
5	1	0	1
4	1	0	0
3	0	1	1
2	0	1	0
1	0	0	1
0	0	0	0

In the above table, if all the pins are open, the get command described above returns the value 7. If pin 2 is shorted to ground (value of 0), the value of the get command is 6 and so forth.

A control array of logic output pins may be specified with the same syntax as in the previous example substituting `digital_gpio_out for digital_gpio_in`.

See Using Events, Logic, and IR and Appendix A: Command Protocol Reference Guide for more information on control array virtual channels.

Understanding IR Receiver Virtual Channel

The IR receiver input on the SoundStructure device responds with acknowledgments when a valid IR signal is received. The first step towards using the IR receiver is to define the IR receiver virtual channel. This can be done with the following syntax:

```
vcdef “IR input” control ir_in 1
```

where 1 is the only physical channel that can be specified since there is only one physical IR receiver channel.

Once a command from the Polycom IR remote transmitter, a command acknowledgment of the form:

```
val ir_key_press “IR Input” 58
```

is generated by the SoundStructure device when a key that corresponds to code 58 is pressed on the IR remote transmitter. The infrared remote controller ID must be set to the factory default of 3 for the IR receiver to properly identify the command.

See Using Events, Logic, and IR for information about how to use the IR receiver with SoundStructure events.

Creating Designs with SoundStructure Studio

A SoundStructure configuration file is a binary file that includes the definition of the virtual channels, the virtual channel groups, the appropriate input and output gain settings, echo cancellation settings, equalization, matrix routings, and more. This file may be uploaded to SoundStructure devices or stored on the local PC for later upload.

By default, SoundStructure products do not have predefined virtual channels or a predefined matrix routing and therefore must be configured before the SoundStructure products can be used in audio applications. The SoundStructure Studio software with integrated InstantDesigner™ is used to create a design and to upload that design to one or more SoundStructure devices.

Note: No Default Configuration for SoundStructure Systems

SoundStructure devices are shipped without a default configuration and must be configured with the SoundStructure Studio software.

The details of creating a new SoundStructure Studio design file are described in this chapter. For information on how to customize a design file, see Customizing SoundStructure Designs and for information on how to use the specific user interface controls with SoundStructure Studio, see Using SoundStructure Studio Controls.

To create a new SoundStructure Studio project, follow these steps:

1. Launch SoundStructure Studio and select New Project from the file menu
2. Follow the on-screen steps to specify the input signals
3. Follow the on-screen steps to specify the output signals
4. Select the SoundStructure devices to be used for the design
5. Create the configuration and optionally upload to the SoundStructure devices

These steps are described in more detail in the following section.

Understanding SoundStructure Studio

The first step to creating a SoundStructure design is to launch the SoundStructure Studio application. If the SoundStructure Studio software is not already installed on the local PC, it may be installed from the CD that was included with the product. More recent versions of SoundStructure Studio may also be available on the Polycom website - please check the Polycom website before installing the SoundStructure Studio version that is on the CD-ROM.

Understanding System Requirements

SoundStructure Studio is supported on Microsoft® Windows XP w/ Service Pack 2 and higher, Microsoft Windows Vista, and Microsoft Windows 7.

SoundStructure Studio requires:

- Microsoft .NET Framework 2.0, which requires 280MB of disk space on an x86 computer architecture, and 610MB on x64 computer architecture
- 40MB of disk space
- 512MB of memory
- A display with 1024x768 resolution.
- A network interface card (wired or wireless) or serial port to connect to SoundStructure devices

Viewing Recommended Operating System

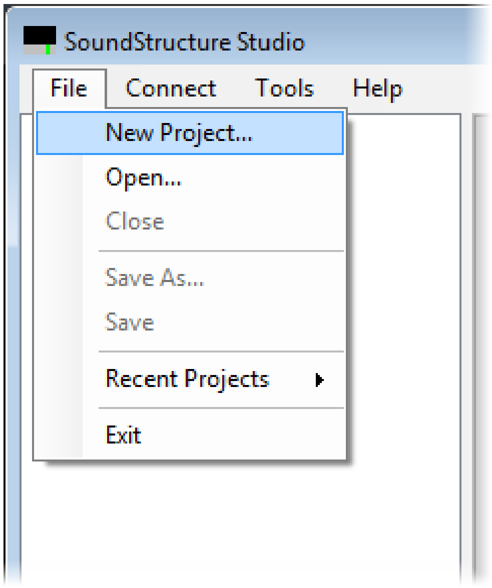
The recommended system for operating SoundStructure Studio has the following characteristics:

- 1GB or higher of memory
- A display with 1280x1024 resolution or higher

Installing SoundStructure Studio

To install SoundStructure Studio,

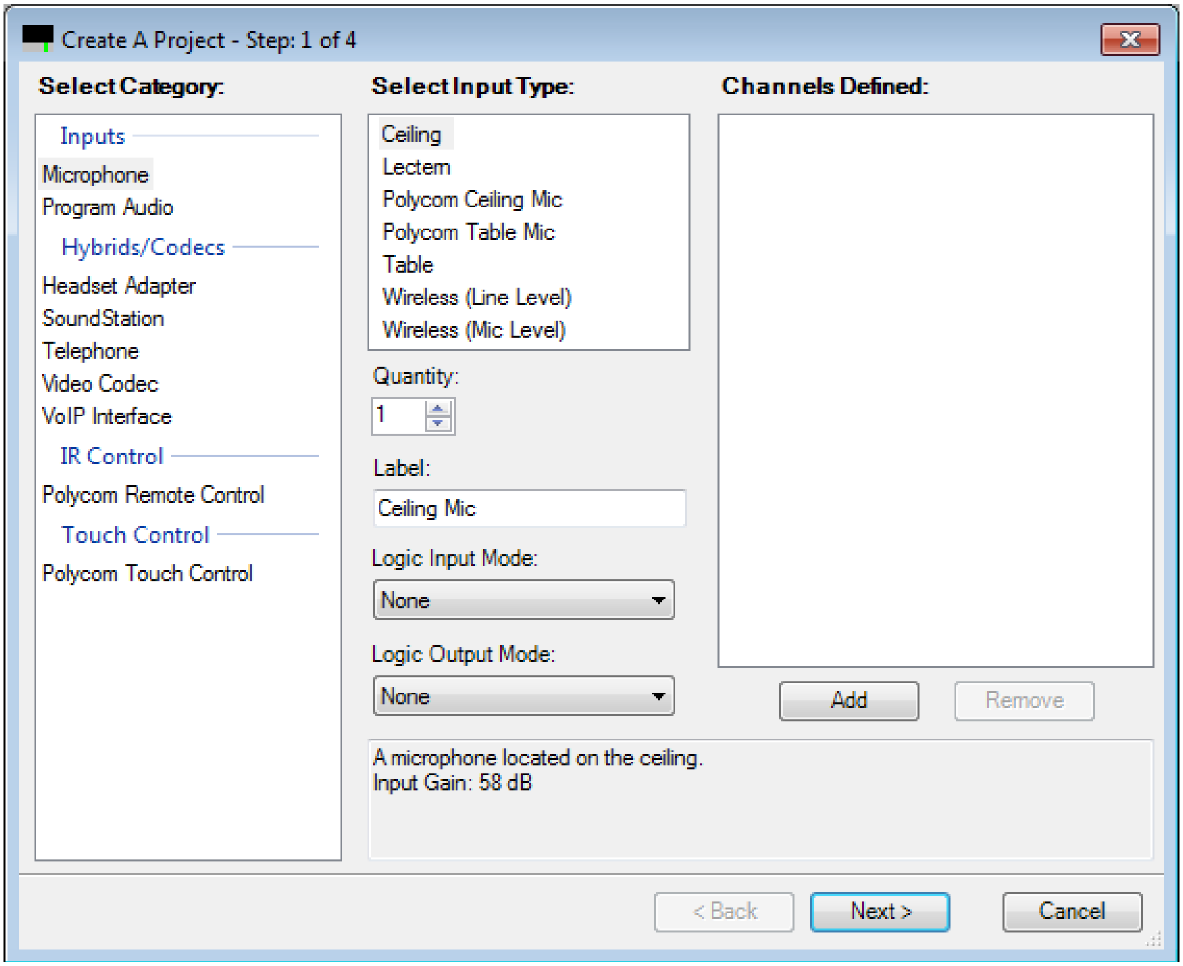
1. Run the StudioSetup.exe software and follow the prompts.
2. After Studio is installed, launch SoundStructure Studio and select **File** > **New Project**, as shown in following figure.



Step 1 - Input Signals

Creating a new project displays the Create a Project window, as shown in the following figure. The first step of the design process is to select the inputs to the system.

Creating A Project Dialog in SoundStructure Studio


Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Headset Adapter
SoundStation
Telephone
Video Codec
VoIP Interface
IR Control
Polycom Remote Control
Touch Control
Polycom Touch Control
Select Input Type:
Ceiling
Lectern
Polycom Ceiling Mic
Polycom Table Mic
Table
Wireless (Line Level)
Wireless (Mic Level)
Channels Defined:
Add
Remove
< Back
Next >
Cancel

Quantity: 1

Label: Ceiling Mic

Logic Input Mode: None

Logic Output Mode: None

A microphone located on the ceiling.
Input Gain: 58 dB

To create a SoundStructure design:

1. Select the style of input (Microphone, Program Audio, etc.), and specify the type of input (Ceiling, Lectern, etc.) and the quantity of the input
2. Click “Add”.

   The label of the input signal becomes the virtual channel name of that input signal. A signal generator is added by default to all projects.

SoundStructure Studio provides a number of predefined input types including microphones, program audio sources, video codecs, telephony interfaces, submixes, and a signal generator.

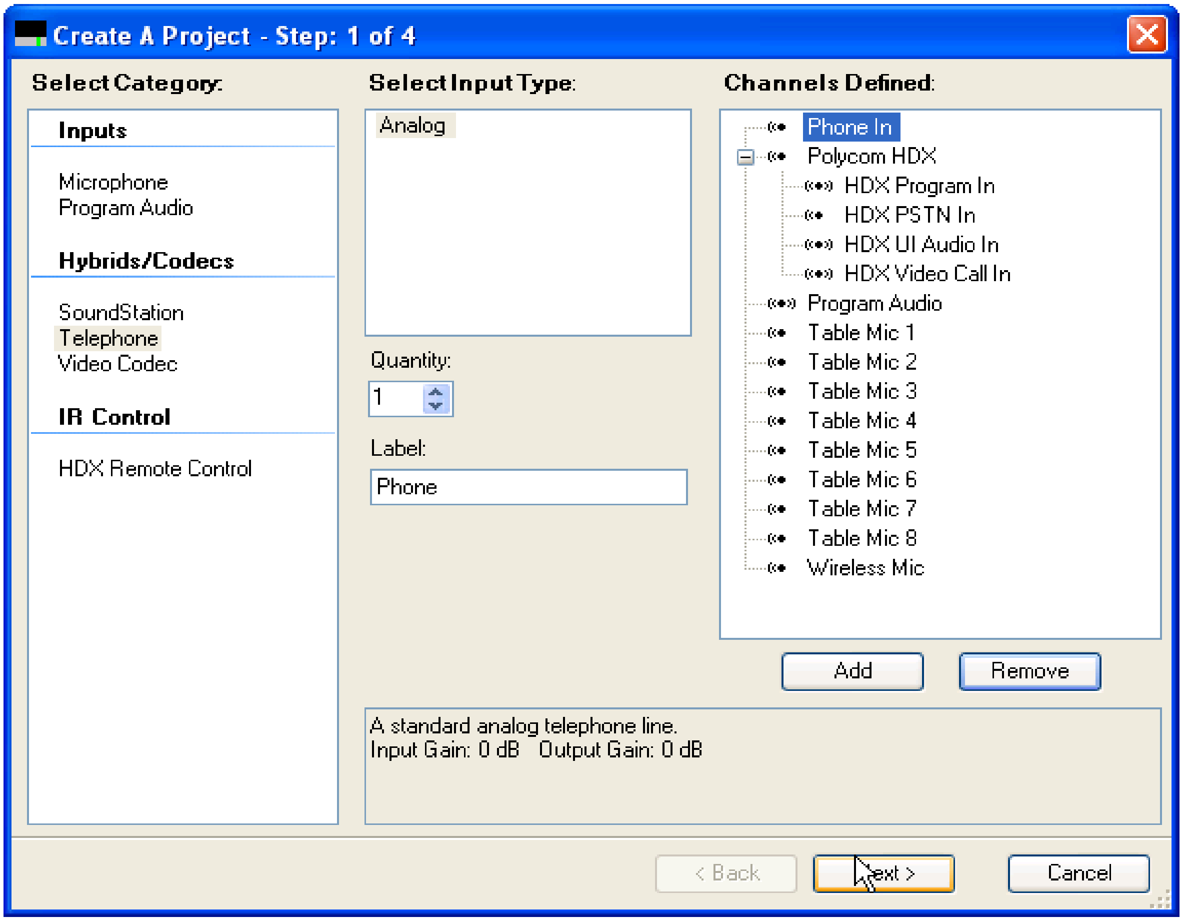
SoundStructure Studio provides default input gains for the various input and output channels. After the design has been created, these gains, along with all other settings, can be adjusted as described in Customizing SoundStructure Designs.

For more information on integration with table and ceiling microphones, see the Best Practices Guide: Polycom SoundStructure and Polycom Microphones.

The choices for Hybrids/Codecs include the Polycom Video Codec, the Polycom VSX series, and a generic mono or stereo video codec. When the Polycom Video Codec is selected, it is assumed that the Polycom Video Codec connects to the SoundStructure device over the Conference Link2 interface. To use the Polycom Video Codec with the SoundStructure devices via the analog input and output instead of Conference Link requires selecting a different codec such as the VSX8000 stereo codec. Connecting Over Conference Link2 provides additional information about integrating with the Polycom Video Codec over the Conference Link2 interface.

A typical system is shown in the next figure where a stereo program audio source, eight table microphones, a wireless microphone, a telephony input, and a Polycom Video Codec have been selected.

Example Project Created in SoundStructure Studio


Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
SoundStation
Telephone
Video Codec
IR Control
HDX Remote Control
Select Input Type:
Analog
Channels Defined:
Phone In
Polycom HDX
HDX Program In
HDX PSTN In
HDX UI Audio In
HDX Video Call In
Program Audio
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
Wireless Mic
Add
Remove
< Back
Next >
Cancel

Quantity: 1

Label: Phone

A standard analog telephone line.
Input Gain: 0 dB   Output Gain: 0 dB

The graphic icon next to the signal name in the Channels Defined: field indicates whether the virtual channel is a monaural channel that is defined with one physical channel (a dot with two waves on one side) or a stereo virtual channel that is defined with two physical channels (a dot with two waves on both sides).

When a Polycom Video Codec is selected, there are multiple audio channels that are created automatically and used independently in the SoundStructure matrix. See Connecting Over Conference Link2 for additional information on the audio channels and the processing that is available on these channels.

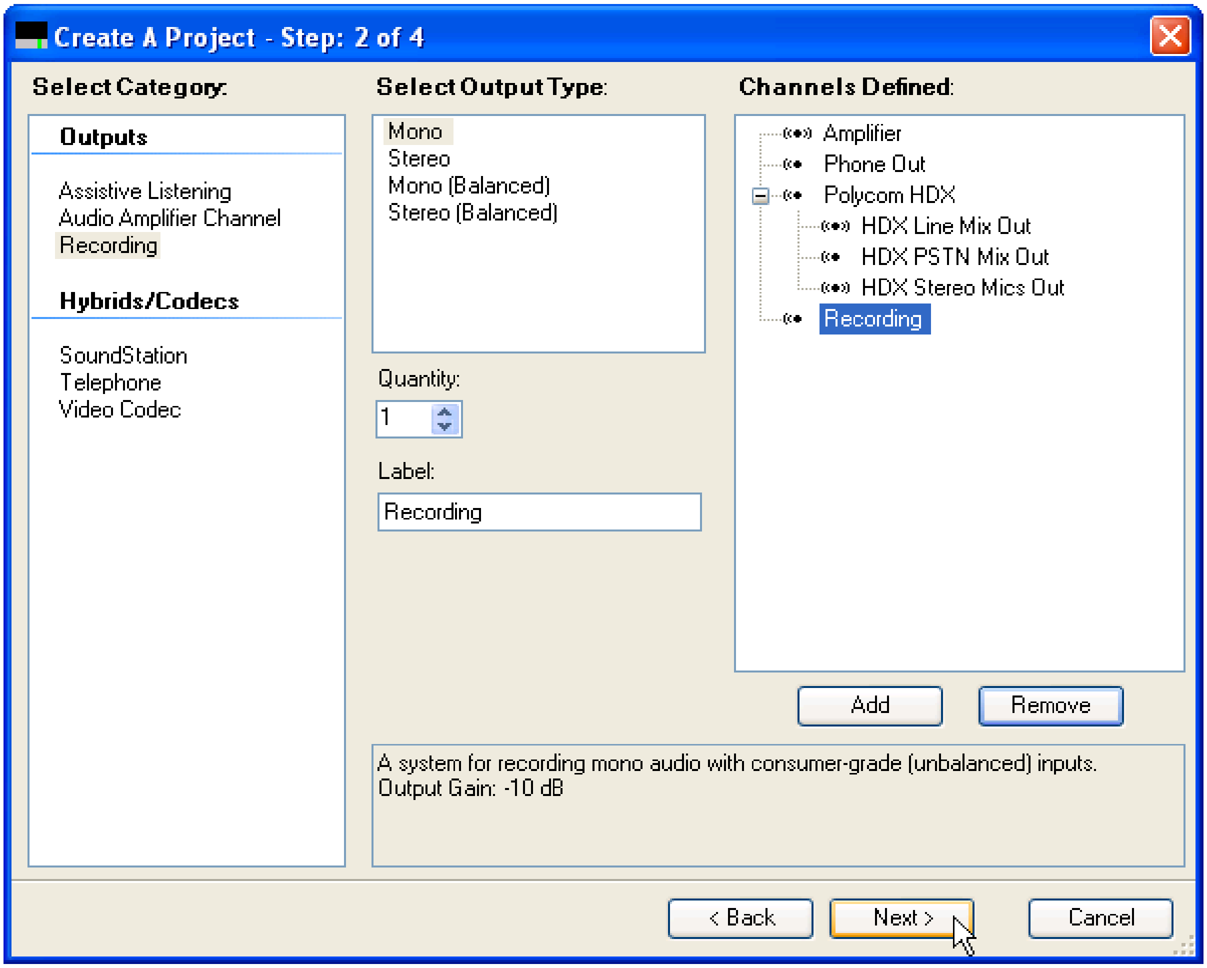
When a video codec or telephony option is selected, the corresponding output signal automatically appears in the outputs page as well.

You can delete Channels by selecting the channel in the Channels Defined: field and clicking Remove.

Step 2 - Output Signals

In step 2 of the design process, the outputs from the system are specified in the same manner that inputs were created. A sample collection of outputs is shown in the following figure.

A Sample Collection of Outputs In SoundStructure Studio


Create A Project - Step: 2 of 4
Select Category:
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
SoundStation
Telephone
Video Codec
Select Output Type:
Mono
Stereo
Mono (Balanced)
Stereo (Balanced)
Channels Defined:
Amplifier
Phone Out
Polycom HDX
HDX Line Mix Out
HDX PSTN Mix Out
HDX Stereo Mics Out
Recording
Add
Remove
< Back
Next >
Cancel

Quantity: 1

Label: Recording

A system for recording mono audio with consumer-grade (unbalanced) inputs.
Output Gain: -10 dB

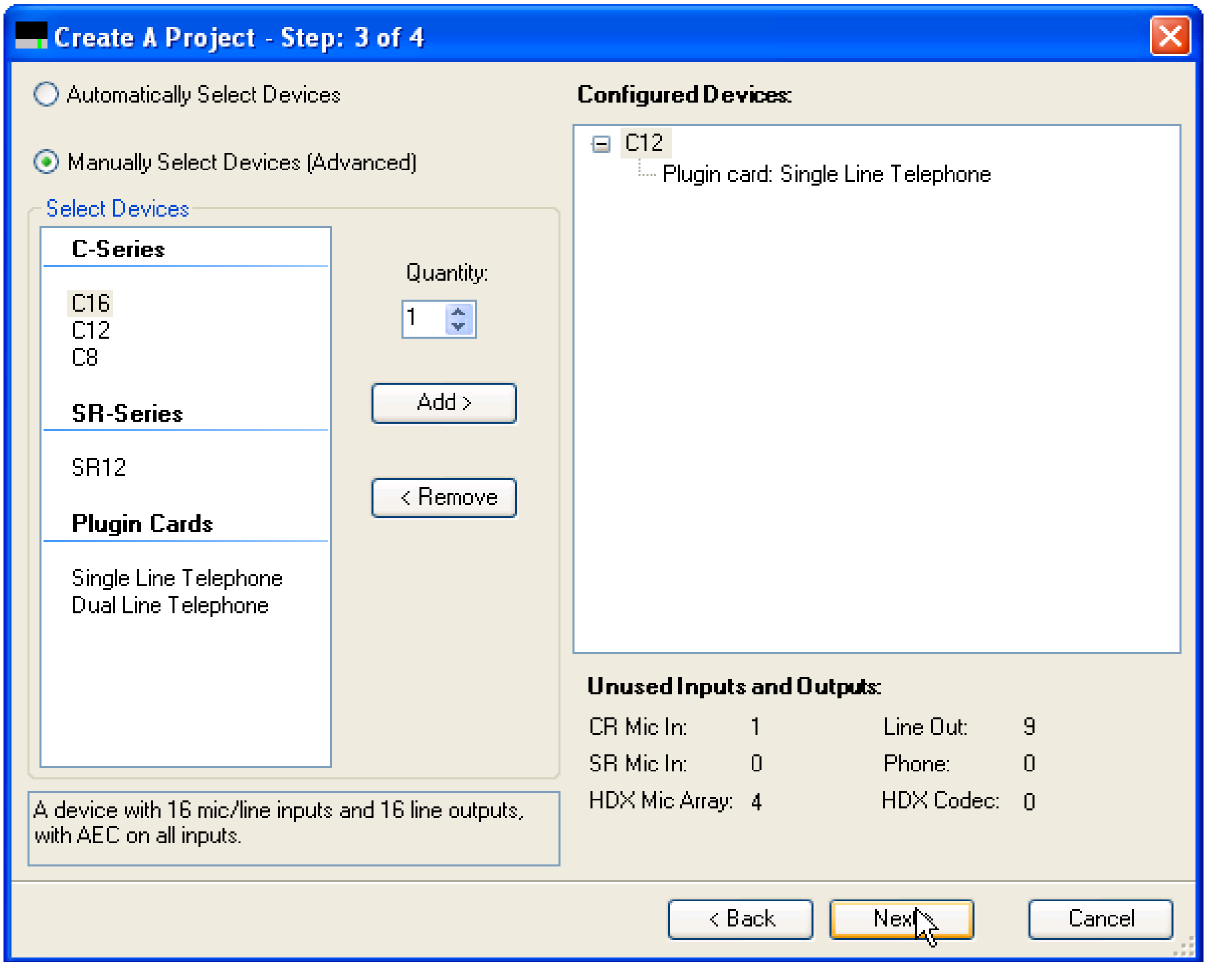
The outputs include audio amplifiers, recording devices, assistive listening devices, and also other telephony or video codec systems. If the desired style of outputs is not found, select something close and then customize the settings as described in Customizing SoundStructure Designs.

In this example, a stereo amplifier was selected as well as a mono recording output. The telephone and Polycom Video Codec conferencing system outputs were automatically created when their respective inputs were added to the system. Notice that there are multiple audio channels associated with the Polycom Video Codec. See Connecting Over Conference Link2 for additional information.

Step 3 - Device Selection

In Step 3, select the devices that you are using with the design project, as shown in the following figure.

Selecting Devices to be Used with a Design Project



Create A Project - Step: 3 of 4

○ Automatically Select Devices

● Manually Select Devices (Advanced)

Quantity: 1

Configured Devices:

- C12
  - Plugin card: Single Line Telephone

A device with 16 mic/line inputs and 16 line outputs, with AEC on all inputs.

Unused Inputs and Outputs:

CR Mic In:	1	Line Out:	9
SR Mic In:	0	Phone:	0
HDX Mic Array:	4	HDX Codec:	0

By default, SoundStructure Studio displays the equipment with the minimum list price although it is possible to manually select the devices by selecting the Manually Select Devices option and adding devices and optional telephony cards.

You can select different devices by clicking on the device, adjusting the quantity, and clicking “Add”. You can remove devices by selecting the device in the **Configured Devices** window and selecting **Remove**.

The unused inputs and outputs display whether additional resources are required to implement the design and also how many unused inputs and outputs are available.

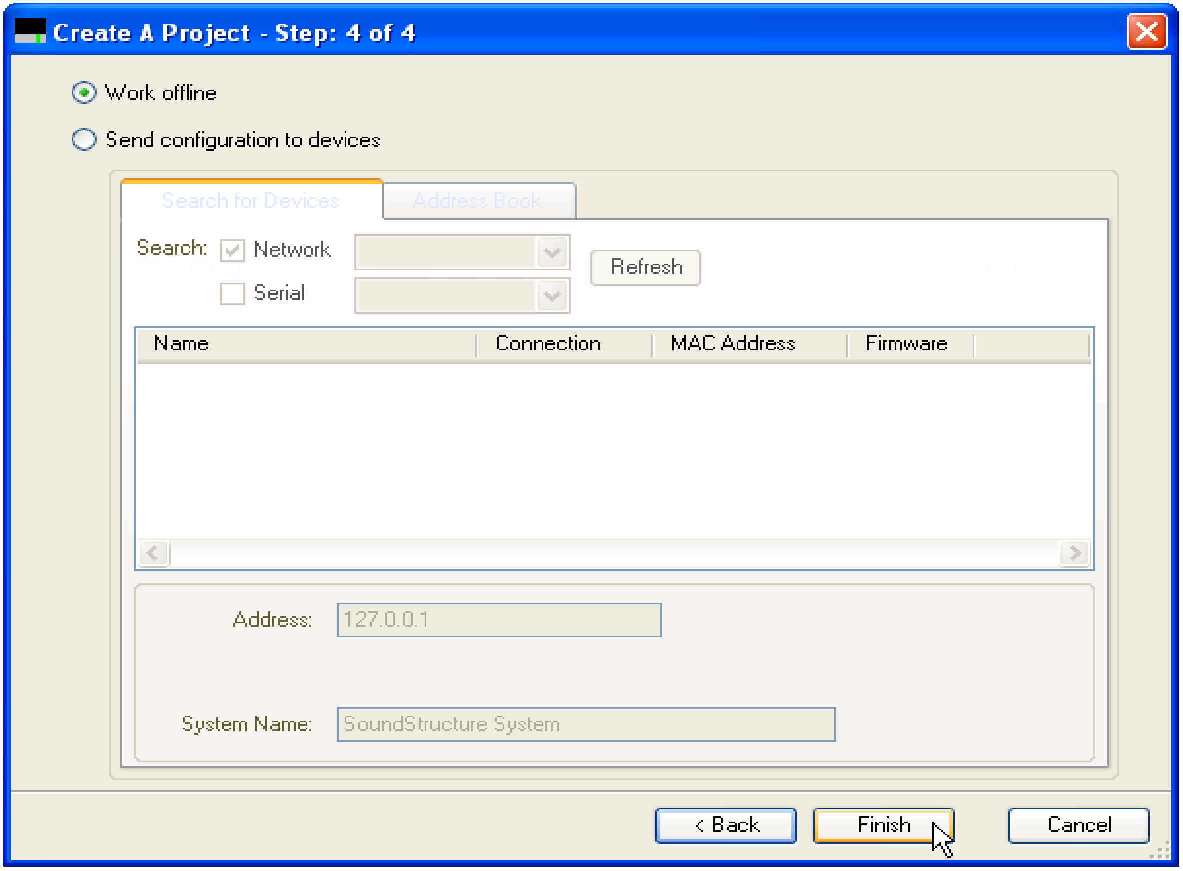
In this example, a SoundStructure C12 and a single-line telephony interface card are selected to implement the design. The resulting system has one additional analog input and nine additional analog outputs. The inputs are used by the eight microphones, one wireless microphone, and the stereo program audio and the line outputs are used by the stereo amplifier and the mono recorder. The Polycom Video Codec does not require any analog inputs and outputs because the signals are transferred over the digital Conference Link2 interface.

Step 4 - Uploading Or Working Offline

In step 4, you can decide to either work offline or work online. When working online, you can select a set of devices to upload the settings to via the Ethernet or RS-232 interfaces. As a best practice, Polycom recommends you design the file offline, customize settings - including the wiring page as described in the Customizing SoundStructure Designs if the system has already been cabled, and upload the settings to the device for final online adjustments.

In this example, the design file is created offline for offline configuration and later uploaded to the device.

Creating Design Projects Offline


Create A Project - Step: 4 of 4

● Work offline

○ Send configuration to devices

Search: ☑ Network ☐ Serial

Address: 127.0.0.1

System Name: SoundStructure System

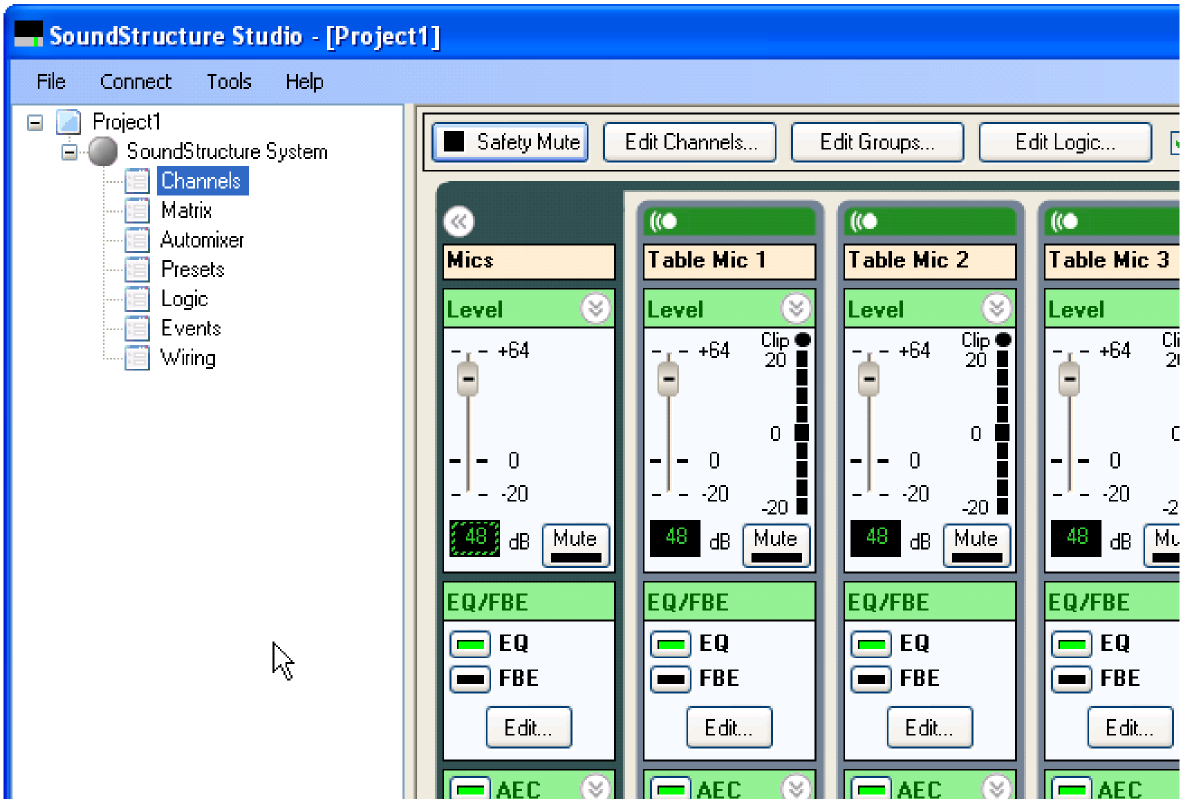
To find devices on the network:

1. Select **Send configuration to devices.**

   SoundStructure Studio searches for devices on the local LAN as defined by the Ethernet interface’s subnet mask or the RS-232 interface. See Installing SoundStructure Devices for additional information on uploading and downloading configuration files and Appendix B: Address Book for how to use the Address Book functionality.

2. Click **Finish**.

   SoundStructure Studio creates a design file including defining all the virtual channels and virtual channel groups such as those shown the following figure.


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring
Safety Mute
Edit Channels...
Edit Groups...
Edit Logic...
Mics
Table Mic 1
Table Mic 2
Table Mic 3
Level
+64
0
-20
Clip
20
48
dB
Mute
EQ/FBE
EQ
FBE
Edit...
AEC

The Customizing SoundStructure Designs describes how to customize the SoundStructure device settings.

If working online, the Ethernet port on the project tree on the left of the screen displays a large green dot next to the device name. When working offline there is a gray dot next to the device name.

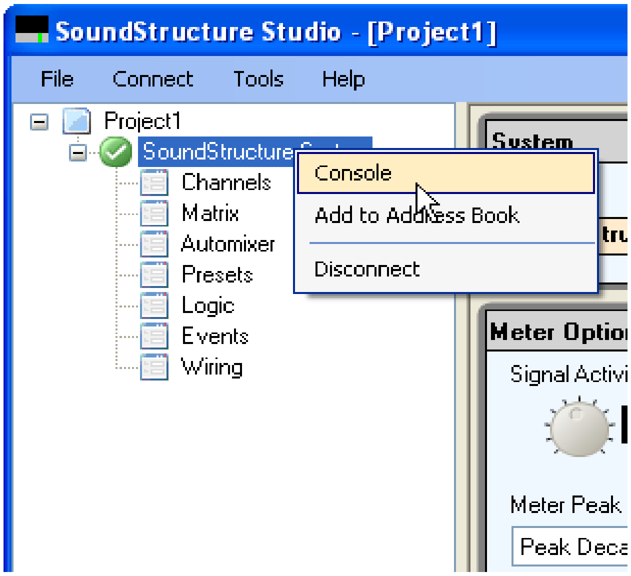
Operating in Online and Offline Mode

SoundStructure Studio has been designed to fully operate in either online or offline modes. Online operation means that SoundStructure Studio is communicating with one or more SoundStructure devices, sending commands to the devices, and receiving command acknowledgments from the devices. Every change to the SoundStructure design is made in real-time to the actual devices. There is no requirement to compile any SoundStructure Studio code before the impact can be heard.

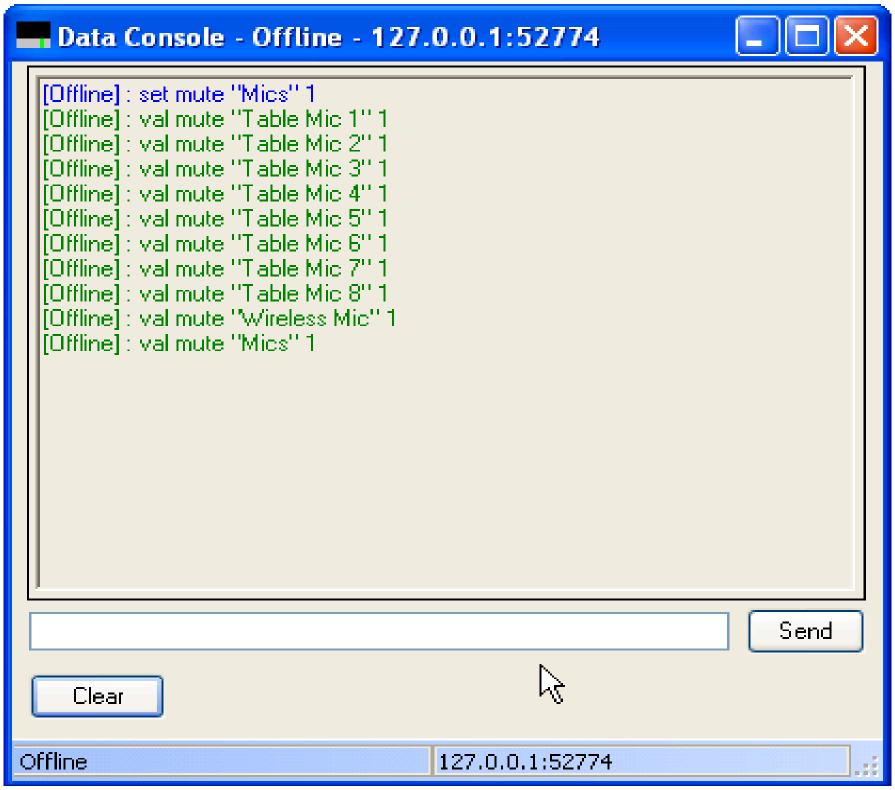
Offline operation means that SoundStructure Studio is working with an emulation of the SoundStructure devices and is not communicating with actual SoundStructure devices. Commands are sent to the emulator and command acknowledgments receive commands from the emulator allowing the designer to test a SoundStructure system design without ever connecting to a system.

Regardless of whether the system is operating online or offline with SoundStructure Studio, you can open the SoundStructure Studio Console and see the commands and acknowledgments by right clicking on the control port interface as shown in the following figures.

SoundStructure Studio Console


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring
Console
Add to Address Book
Disconnect
System
Meter Option
Signal Activi
Meter Peak
Peak Deca

SoundStructure Studio Data Console


Data Console - Offline - 127.0.0.1:52774
Send
Clear
Offline
127.0.0.1:52774

```
[Offline] : set mute "Mics" 1
[Offline] : val mute "Table Mic 1" 1
[Offline] : val mute "Table Mic 2" 1
[Offline] : val mute "Table Mic 3" 1
[Offline] : val mute "Table Mic 4" 1
[Offline] : val mute "Table Mic 5" 1
[Offline] : val mute "Table Mic 6" 1
[Offline] : val mute "Table Mic 7" 1
[Offline] : val mute "Table Mic 8" 1
[Offline] : val mute "Wireless Mic" 1
[Offline] : val mute "Mics" 1
```

In this example, the virtual channel group “Mics” are muted and the console shows the command in blue and the acknowledgments generated in green.

When SoundStructure Studio is working offline, the prefix **[Offline]:** is shown in the console as a reminder that commands are not being sent to actual devices. While offline, commands are sent to the SoundStructure device emulator using the command syntax described in Appendix A: Command Protocol Reference Guide and acknowledgments are received just as if communicating to actual systems.

Offline operation is commonly used prior to the actual installation of the physical SoundStructure devices to adjust the system before on site installation, or when a physical device is not readily accessible.

Note: Working Offline with SoundStructure Studio

With SoundStructure Studio, it is possible to work offline and fully emulate the operation of the SoundStructure devices. You can send commands to the system, the system receives acknowledgments, and the system operation including presets, signal gains, matrix crosspoints, and more are tested without ever connecting to SoundStructure devices.

When working offline, you can save the configuration file at any time by selecting **File** > **Save Project**. This creates the file with the name of your choosing and stores the file on the local disk with the *.str* file extension.

When working online, saving the project prompts you to save the file on the disk as well as store the settings in the SoundStructure device.

Customizing SoundStructure Designs

After you create a SoundStructure project file as described in Creating Designs with SoundStructure Studio, you can use the SoundStructure Studio software to adjust and customize the design. This section provides you with in-depth instructions on how to customize the settings by using the Wiring, Channels, Matrix, Telephony, and Automixer pages. For information on uploading and downloading configuration files, see Installing SoundStructure Devices.

The detailed controls for the inputs, outputs, and submix signals are presented in the order that the controls appear on the channels page.

After you make changes to the configuration, ensure that the settings are stored to a preset (see Installing SoundStructure Devices) and that you define a power on preset.

Using the Wiring Page

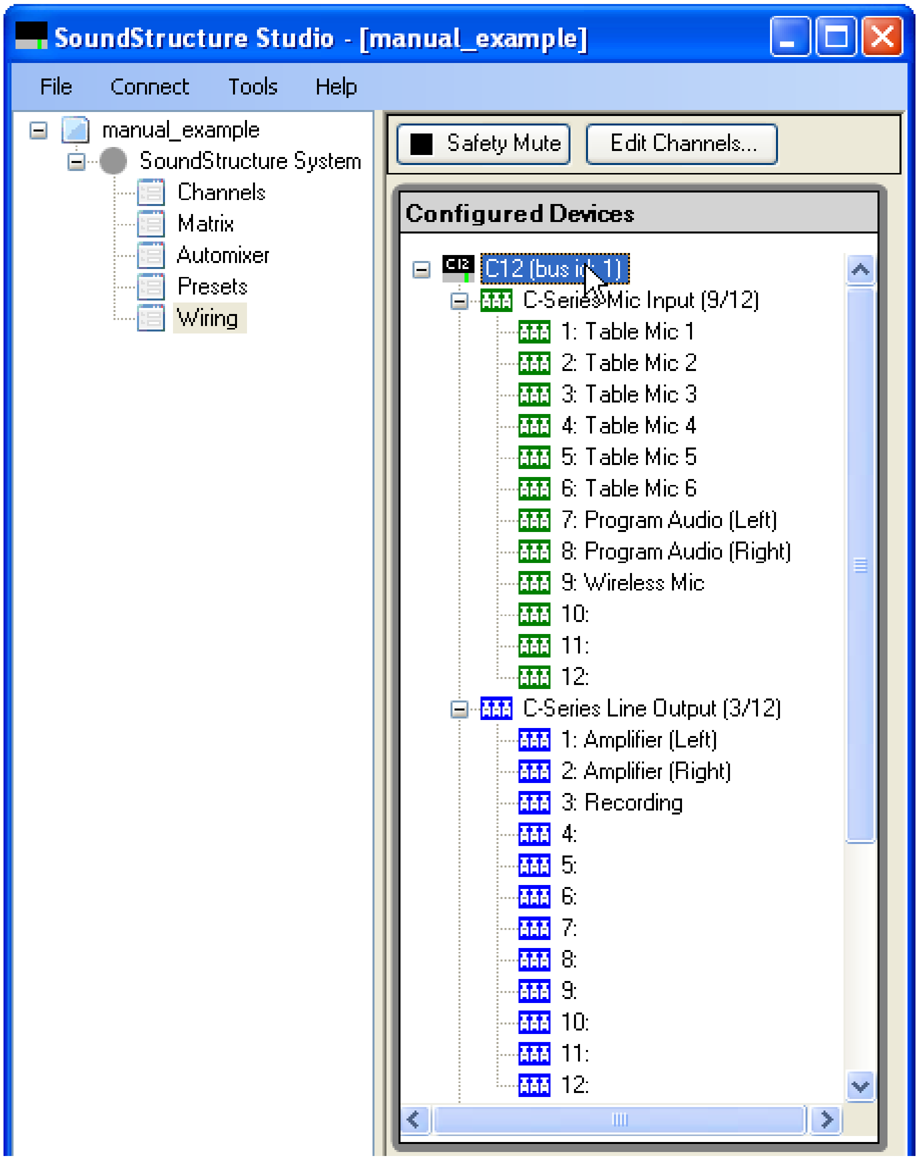
During the design process, SoundStructure Studio creates the virtual input and output channels using the labels that were used during design steps 1 and 2 in Creating Designs with SoundStructure Studio as the virtual channel names. The virtual channels are created with default physical input and output channels which are assigned automatically based on the order that the virtual channels are added to the system during the first two design steps. Changing the order that inputs and outputs are selected changes the default physical channel assignments.

The wiring page is where the SoundStructure Studio wiring assignment are reviewed and changed if SoundStructure Studio wired the system with different inputs and outputs than expected or desired.

As shown in the example in the following figure, the six table top microphones use physical inputs 1 - 6, the program audio uses inputs 7 and 8 and the wireless microphone uses input 9. On the outputs, the amplifier stereo virtual channel uses physical channels 1 and 2 and the recording channel uses physical output 3. Remember that stereo virtual channels are always defined with two physical channels while mono virtual channels are defined with one physical channel.

The following figure shows the default wiring for an example that the system created with six table top microphones, stereo program audio, and a wireless microphone.

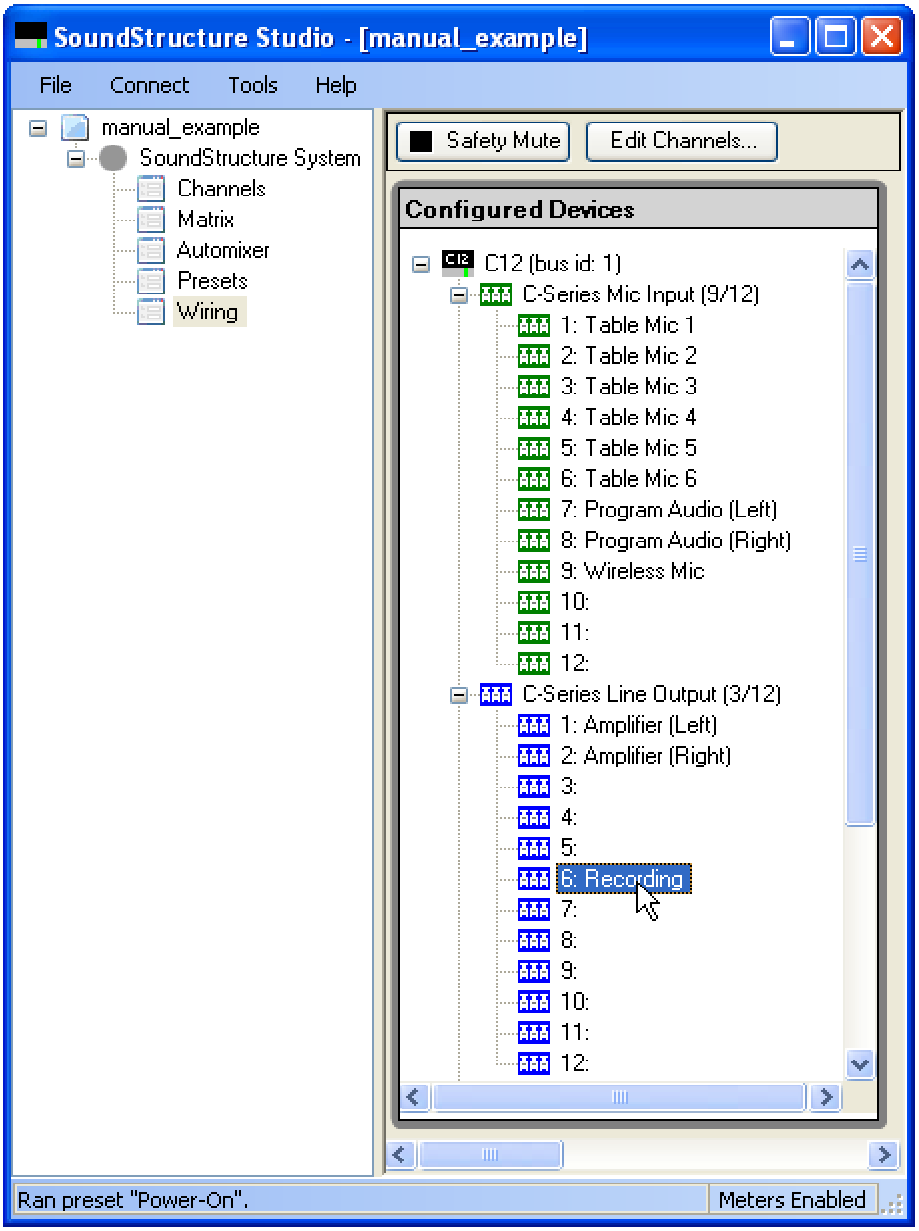
An Example SoundStructure Device with Default Wiring


SoundStructure Studio - [manual_example]
File
Connect
Tools
Help
manual_example
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Configured Devices

C12 (bus id: 1)
- C-Series Mic Input (9/12)
  - 1: Table Mic 1
  - 2: Table Mic 2
  - 3: Table Mic 3
  - 4: Table Mic 4
  - 5: Table Mic 5
  - 6: Table Mic 6
  - 7: Program Audio (Left)
  - 8: Program Audio (Right)
  - 9: Wireless Mic
  - 10:
  - 11:
  - 12:
- C-Series Line Output (3/12)
  - 1: Amplifier (Left)
  - 2: Amplifier (Right)
  - 3: Recording
  - 4:
  - 5:
  - 6:
  - 7:
  - 8:
  - 9:
  - 10:
  - 11:
  - 12:

If it is necessary to change the wiring from the default wiring, you can change the virtual wiring by clicking and dragging signals from their current input or output to a new input or output, as shown in the following figure. In this example, the Recording output changed from physical output 3 to physical output 6.

Editing Default Wiring in SoundStructure Studio


SoundStructure Studio - [manual_example]
File
Connect
Tools
Help
manual_example
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Configured Devices
C12 (bus id: 1)
C-Series Mic Input (9/12)
1: Table Mic 1
2: Table Mic 2
3: Table Mic 3
4: Table Mic 4
5: Table Mic 5
6: Table Mic 6
7: Program Audio (Left)
8: Program Audio (Right)
9: Wireless Mic
10:
11:
12:
C-Series Line Output (3/12)
1: Amplifier (Left)
2: Amplifier (Right)
3:
4:
5:
6: Recording
7:
8:
9:
10:
11:
12:
Meters Enabled

Ran preset "Power-On".

When a virtual channel is moved, SoundStructure Studio redefines the virtual channel to use the new physical inputs or outputs that are specified. Moving a virtual channel does not create any visible changes in the Matrix or Channels page because SoundStructure Studio operates at the level of the virtual channel and not the physical channels. The only page that displays a difference is the Wiring page.

It is important to know that the actual wiring of the system needs to match the wiring specified on the Wiring page. Otherwise, the system does not operate as expected. For instance, in the example above, if the recording output is physically plugged into output 3 when SoundStructure Studio notices the recording output is plugged into output 6, no audio is heard on output 3 because the audio is being routed to physical output 6.

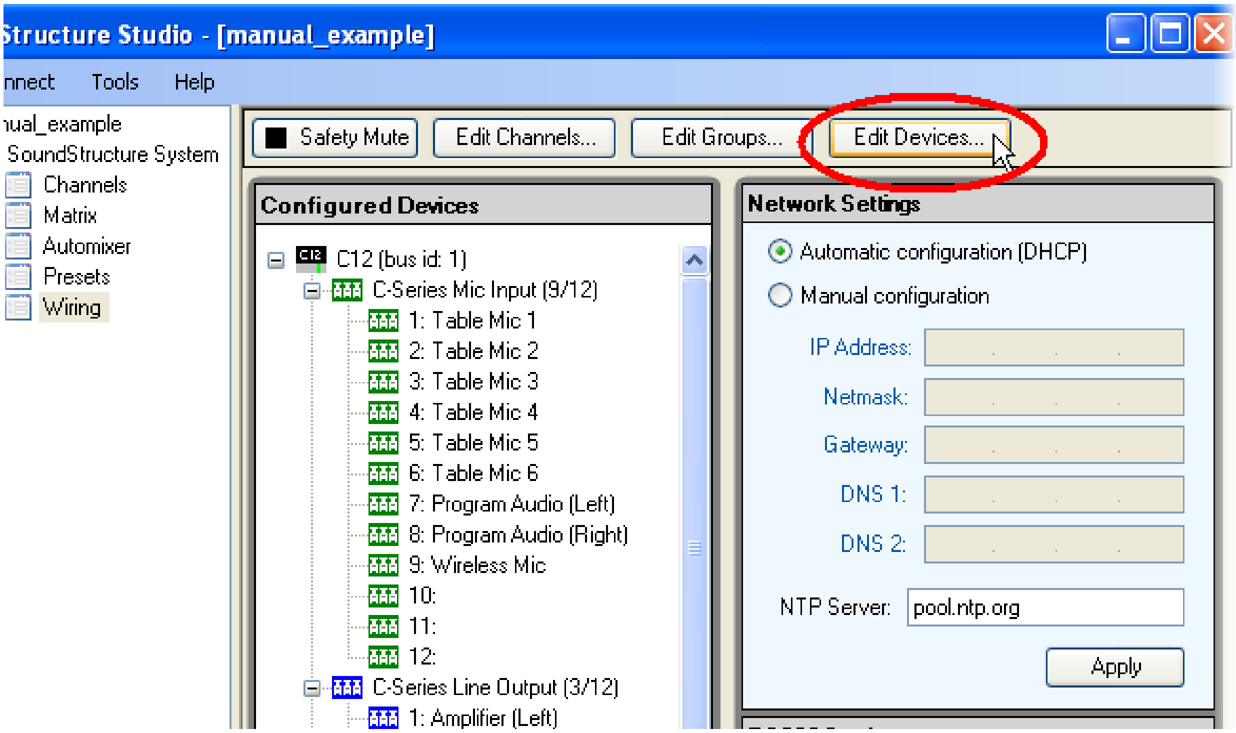
Note: Matching Physical Channel Wiring

For proper system operation, make sure the physical channel wiring matches the wiring instructions on the Channel page. You can make adjustments to the wiring by physically moving connections to match the Wiring page, or by moving signals on the Wiring page to match the physical connections.

Editing Devices

When working offline, the Wiring Page includes an **Edit Devices** control for changing the underlying SoundStructure equipment that was selected during the design process, as shown in the following figure.

Edit Devices in SoundStructure Studio



Structure Studio - [manual_example]
nnect
Tools
Help
ual_example
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Edit Groups...
Edit Devices...
Configured Devices
Network Settings
Apply

- C12 (bus id: 1)
  - C-Series Mic Input (9/12)
    - 1: Table Mic 1
    - 2: Table Mic 2
    - 3: Table Mic 3
    - 4: Table Mic 4
    - 5: Table Mic 5
    - 6: Table Mic 6
    - 7: Program Audio (Left)
    - 8: Program Audio (Right)
    - 9: Wireless Mic
    - 10:
    - 11:
    - 12:
  - C-Series Line Output (3/12)
    - 1: Amplifier (Left)

● Automatic configuration (DHCP)

○ Manual configuration

IP Address: . . .

Netmask: . . .

Gateway: . . .

DNS 1: . . .

DNS 2: . . .

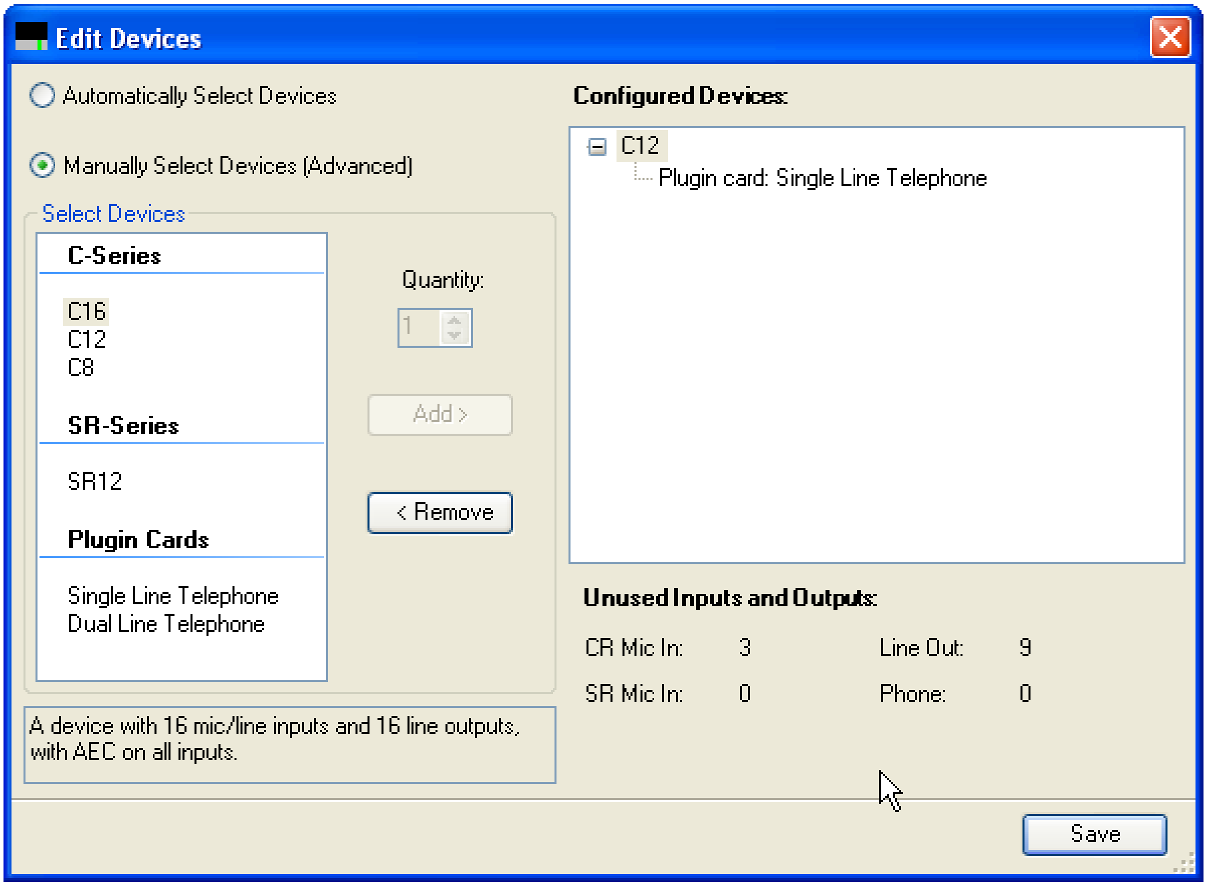
NTP Server: pool.ntp.org

You can do the following with the Edit Devices control:

- Grow a project from a smaller SoundStructure device to a larger device
- Shrink a project from a larger SoundStructure device to a smaller device, if there are enough unused inputs and outputs
- Add, change, or remove telephony cards

The Edit Devices control that displays is the same control that was used during the original design process and is shown below.

Edit Devices Page in SoundStructure Studio



○ Automatically Select Devices

● Manually Select Devices (Advanced)

C-Series

- C16
- C12
- C8

SR-Series

- SR12

Plugin Cards

- Single Line Telephone
- Dual Line Telephone

A device with 16 mic/line inputs and 16 line outputs, with AEC on all inputs.

Configured Devices:

- C12
  - Plugin card: Single Line Telephone

Unused Inputs and Outputs:

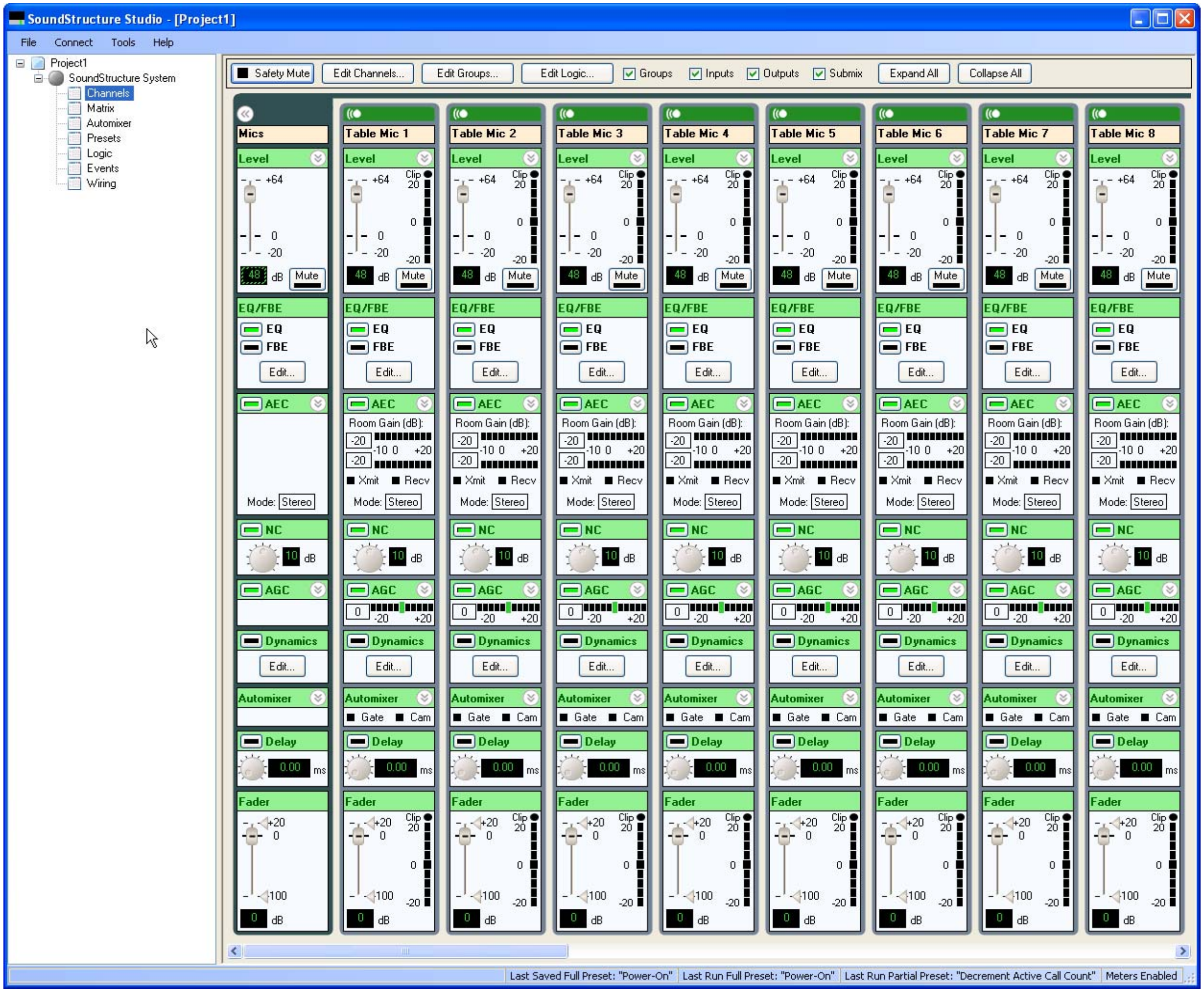
CR Mic In:	3	Line Out:	9
SR Mic In:	0	Phone:	0

To reduce the equipment on a project that has too many inputs or outputs to fit into the next smaller SoundStructure device requires removing audio channels from the Edit Channel control.

Using the Channels Page

The Channels page is the primary area for customizing the signal gains and processing for the input, output, and submix signals. Regardless of the number of SoundStructure devices used in a design, there is only one Channels page and that page displays all the virtual channels for the entire design. A typical Channels page is shown in the following figure.

Channels Page in SoundStructure Studio


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring
Safety Mute
Edit Channels...
Edit Groups...
Edit Logic...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Expand All
Collapse All
Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
Level
+64
0
-20
Clip
20
48
dB
Mute
EQ/FBE
EQ
FBE
Edit...
AEC
Room Gain (dB):
-20
-10
+20
Xmit
Recv
Mode: Stereo
NC
10
dB
AGC
Dynamics
Automixer
Gate
Cam
Delay
0.00
ms
Fader
+20
-100
0

Last Saved Full Preset: "Power-On" | Last Run Full Preset: "Power-On" | Last Run Partial Preset: "Decrement Active Call Count" | Meters Enabled

The input and output signals are shown with different colored outlines to differentiate among inputs, outputs, and submixes. The signals are color coded with the input signals having a green shading and outline, the output signals having a blue shading and outline to match the rear-panel labeling, and the submixes have a purple shading and outline, as shown in the following figure.

Color Coding for Inputs, Outputs, and Submixes on the Channels Page in SoundStructure Studio

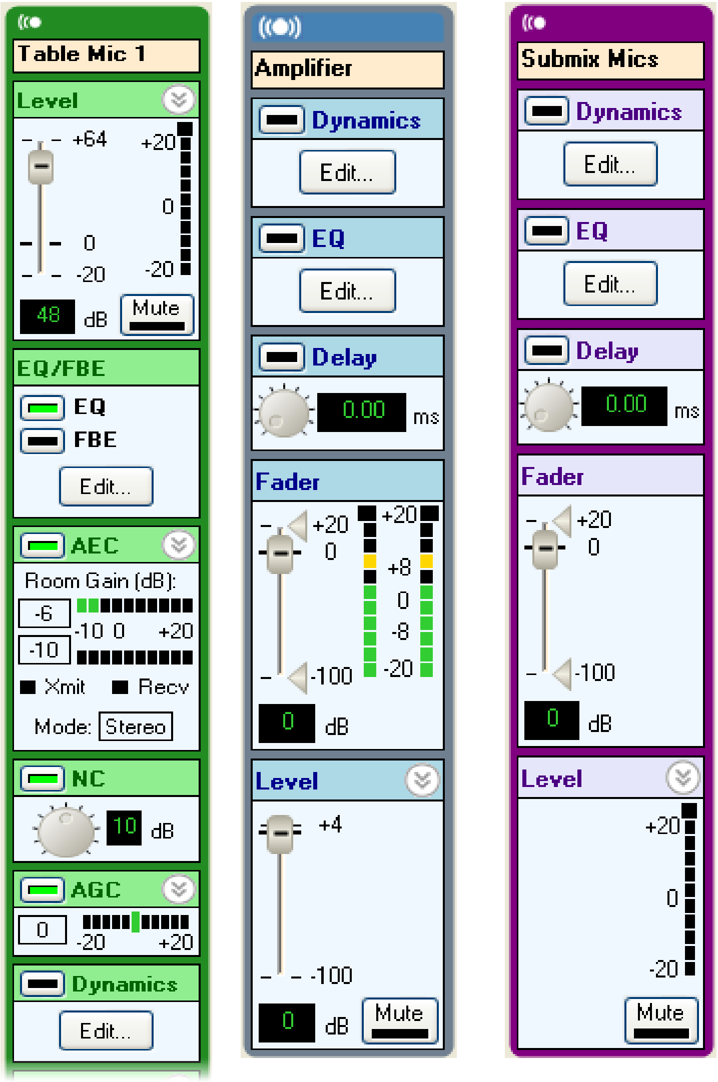
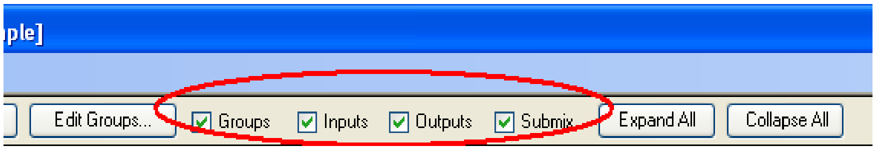


Table Mic 1
Level
+64
+20
0
-20
48 dB
Mute
EQ/FBE
EQ
FBE
Edit...
AEC
Room Gain (dB):
-6
-10 0 +20
-10
Xmit
Recv
Mode: Stereo
NC
10 dB
AGC
0
-20 +20
Dynamics
Edit...
Amplifier
Dynamics
Edit...
EQ
Edit...
Delay
0.00 ms
Fader
+20
0
+8
-8
-100
-20
0 dB
Level
+4
-100
0 dB
Mute
Submix Mics
Dynamics
Edit...
EQ
Edit...
Delay
0.00 ms
Fader
+20
0
-100
0 dB
Level
+20
0
-20
Mute

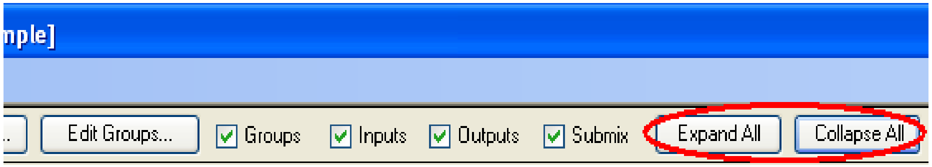
You can change which types of virtual channels are viewed by enabling or disabling groups, inputs, outputs, and submixes with the controls on the top of the Channels page as shown in the following figure.

Editing Controls on the Channels Page in SoundStructure Studio


ple]
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Expand All
Collapse All

In addition, you can expand groups of virtual channels to display the individual members of the group by clicking **Expand All** or collapse the channels to only show the virtual channel groups by clicking **Collapse All**, as shown in the following figure.

Editing Controls on the Channels Page in SoundStructure Studio


mple]
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Expand All
Collapse All

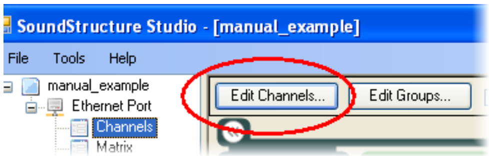
Note: Adjusting Virtual Channel Settings

Any of the settings for virtual channels can be adjusted by either adjusting the virtual channels individually or by adjusting the virtual channel group settings.

Editing Virtual Channels

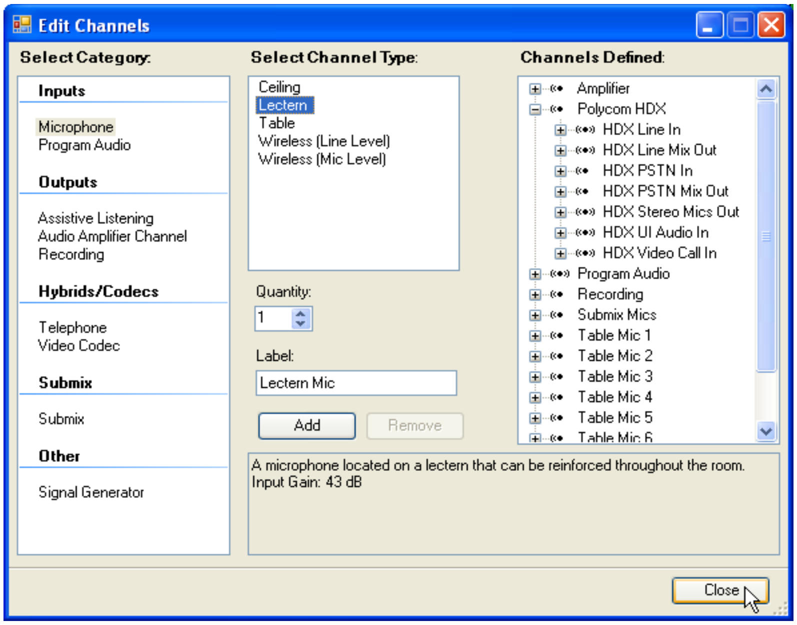
You can add or delete additional virtual channels by clicking **Edit Channels** on the Channels page as highlighted in the following figure. You can adjust designs to add more inputs or outputs up to the limit of the number of physical inputs and outputs of the hardware that was selected to implement the design.

Editing Channels on the Channels Page in SoundStructure Studio


SoundStructure Studio - [manual_example]
File
Tools
Help
manual_example
Ethernet Port
Channels
Matrix
Edit Channels...
Edit Groups...

The Edit Channels button opens the input and output channel selection window and enables you to add or remove virtual channels, as shown in the following figure. If virtual channels are added, the channels display on the Channels page with default gain settings for the devices and default signal routing created for the matrix based on the type of signal added. If virtual channels are deleted, the channels are removed from the Channels page and the channels' matrix signal routings are removed.

The Edit Channels Page in SoundStructure Studio


Edit Channels
Select Category:
Inputs
Microphone
Program Audio
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Telephone
Video Codec
Submix
Submix
Other
Signal Generator
Select Channel Type:
Ceiling
Lectern
Table
Wireless (Line Level)
Wireless (Mic Level)
Channels Defined:
Amplifier
Polycom HDX
HDX Line In
HDX Line Mix Out
HDX PSTN In
HDX PSTN Mix Out
HDX Stereo Mics Out
HDX UI Audio In
HDX Video Call In
Program Audio
Recording
Submix Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Add
Remove
Close

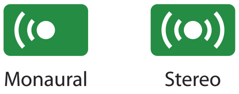
Quantity: 1

Label: Lectern Mic

A microphone located on a lectern that can be reinforced throughout the room.
Input Gain: 43 dB

There is a graphic symbol, see the following figure, at the top of each virtual channel as a reminder of whether the virtual channel is a monaural or stereo virtual channel.

Monaural and Stereo Virtual Channel Symbols


Monaural
Stereo

This graphic symbol is also shown on the Edit Channels page associated with each channel in the 'Channels Defined:' column.

Creating Virtual Channel Groups

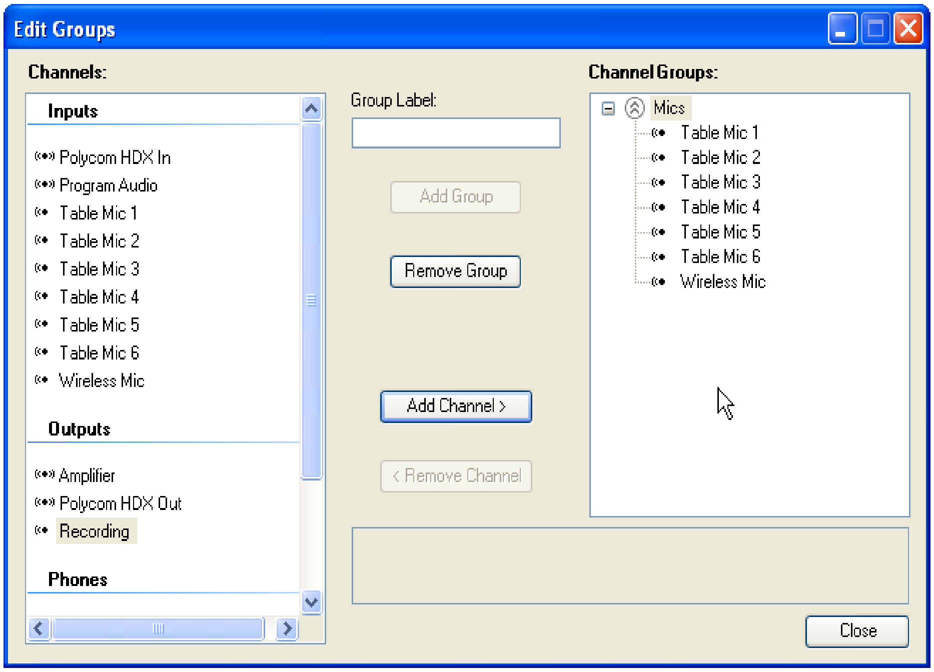
Virtual channel groups are collections of virtual channels that you can configure together. When creating a new project, a virtual channel group called "Mics" is automatically created and includes all the microphone inputs for the design. The virtual channel group can be used to adjust all the settings for all the signals in the virtual group regardless of whether the group is expanded or contracted.

A virtual channel group may be collapsed or expanded by clicking the « » graphics respectively, on the top of the group page. All groups in the channels page can be expanded or collapsed by clicking on the Expand or Collapse buttons respectively.

To create additional virtual channel groups:

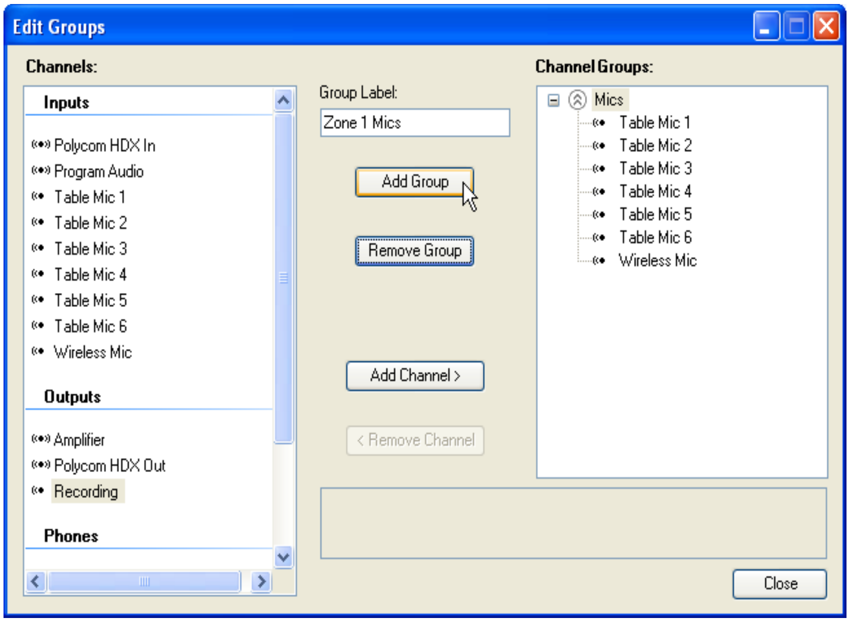
» Click **Edit Groups** on the Channels page.

All existing virtual channel groups display on the right of the screen. Virtual channels can be in more than one virtual channel group. For example, Table Mic 1 can be in the virtual channel group Mics and Zone 1 Mics.


Edit Groups
Channels:
Inputs
Polycom HDX In
Program Audio
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Wireless Mic
Outputs
Amplifier
Polycom HDX Out
Recording
Phones
Group Label:
Add Group
Remove Group
Add Channel >
< Remove Channel
Channel Groups:
Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Wireless Mic
Close

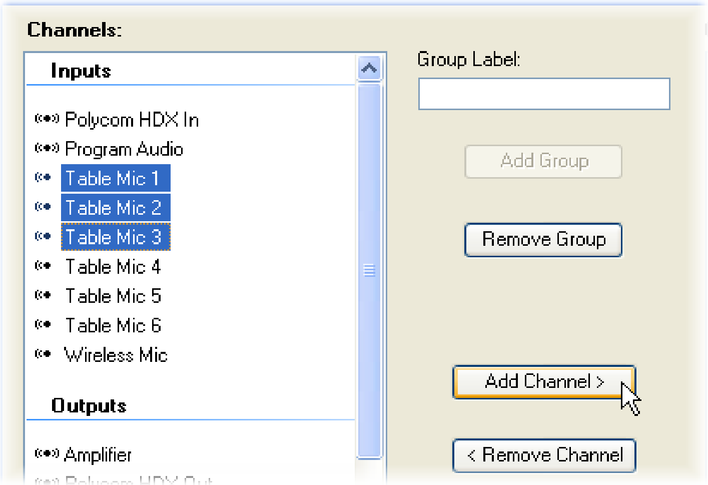
To add a new virtual channel group:

» Enter a group name in the **Group Label:** field and click **Add Group,** as shown in the following figures. This figure shows an example of creating the Zone 1 Mics virtual channel group.


Edit Groups
Channels:
Inputs
Polycom HDX In
Program Audio
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Wireless Mic
Outputs
Amplifier
Polycom HDX Out
Recording
Phones
Add Group
Remove Group
Add Channel >
< Remove Channel
Channel Groups:
Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Wireless Mic
Close

Group Label: Zone 1 Mics

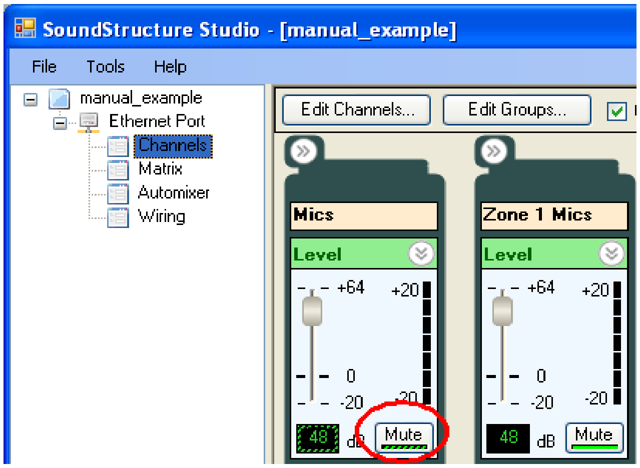
After you define a virtual channel group, you can add virtual channels to the virtual channel group by selecting the desired virtual channels. You can select more than one virtual channel by left clicking on the first channel and holding shift while you click on subsequent virtual channels. After you select the virtual channels, click **Add Channel**, as shown in the following figure.


Channels:
Inputs
Polycom HDX In
Program Audio
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Wireless Mic
Outputs
Amplifier
Group Label:
Add Group
Remove Group
Add Channel >
< Remove Channel

Any commands sent to configure the virtual channel group are sent to the members of the virtual channel group. For example, if a mute command is sent to Zone 1 Mics then Table Mic 1, Table Mic 2, and Table Mic 3 are all muted and the Zone 1 Mics logical group display as muted.

If individual members of a group have different values for the same parameter, such as the mute state, the value of the group parameter displays with a crosshatch pattern, as shown in the following figures.

Virtual Channels Muted



If the Mics group is unmuted and the Zone 1 Mics group is muted, the mute status of the Zone 1 Mics displays the mute status and the Mics group displays a mixed mute state because some microphones in the group are still muted but others are unmuted. The mixed mute state is shown as a cross hatched bar in the mute button.

Notice in the above figure that the gain for the microphone inputs in the Mics group displays as 43 with dashed lines around it indicates that some - but not all - of the microphones have a gain of 43 dB. In this example, the wireless microphone has a different gain value. The group displays a dashed line if all the values are not the same for the members in the group. In the above figure, all the members of the Zone 1 Mics group have 48 dB of gain, so there are no dashed lines around the gain for the Zone 1 Mics group.

Note: Changing Virtual Channels and Groups

Changing virtual channel group settings changes all the settings for the virtual channels that are a part of the virtual channel group and generate command acknowledgments for the virtual channel group and its virtual channels members.

If a parameter for all members of a virtual channel group is individually changed to the same value, the virtual channel group setting does not set automatically to the common value and consequently are no command acknowledgment that the virtual channel group has that common value. For instance, if all microphones in the Zone 1 group are muted individually, the Zone 1 group does not acknowledge that the group is muted. However, if the Zone 1 group is muted, Zone 1 group acknowledges that the group and all the members of the group are in a muted state.

Note: Individually Changing Members in a Virtual Channel Group

Changing the settings of all members in the group individually to a common value does not cause the virtual channel group to show that common value.

Setting Input Signals

The settings applied to input signals depend on the type of virtual channel created from that physical input. For example, there are different controls if the signal is a microphone input, line level input, a stereo virtual channel, a signal generator, or a telco input.

Enabling Input Signal Meters

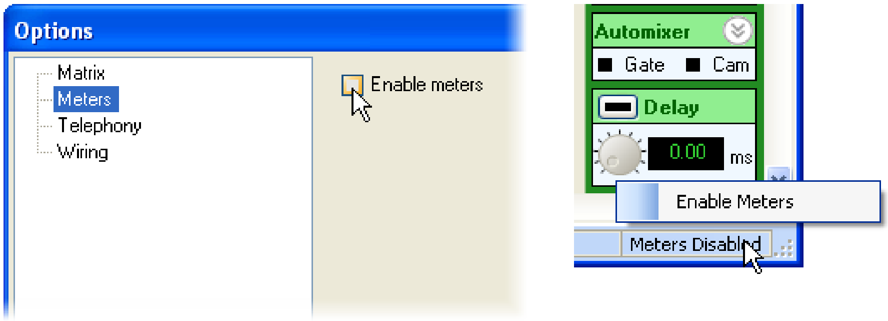
All input signals have meters that display the signal activity. The meters are enabled from the Tools menu or from the lower right hand corner of the screen.

To enable the signal meters from the Tools menu:

1 Select **Tools** > **Options**.

2 Choose the meters entry and select **Enable Meters**.

You can also enable meters by right clicking on the lower right hand corner of the screen and select the desired meter state. Both options are shown in the following figure.



Enabling meters is a function of SoundStructure Studio and not the particular configuration file. This means that when you enable meters, the meters are enabled for all projects that SoundStructure Studio opens from then on.

After you enable meters and navigate to a page that displays the meter activity (such as the Channels page), the desired signal meters are automatically registered by SoundStructure Studio and the meter data is sent from the SoundStructure device to SoundStructure Studio. Navigating away from a page with meter information causes the meters unregister and any new meters on the new page are registered. SoundStructure Studio uses the `mtrreg` and `mtrunreg` commands to automatically register and unregister meters, respectively

You can view meter information either over RS-232 or Ethernet connections to the SoundStructure device; however, the meters are most responsive over a Ethernet connection. If meters are viewed over the RS-232 interface, Polycom recommends that you use the highest data rate of 115,200 baud to minimize any lag between registering for meters and having the meter information displayed on the screen.

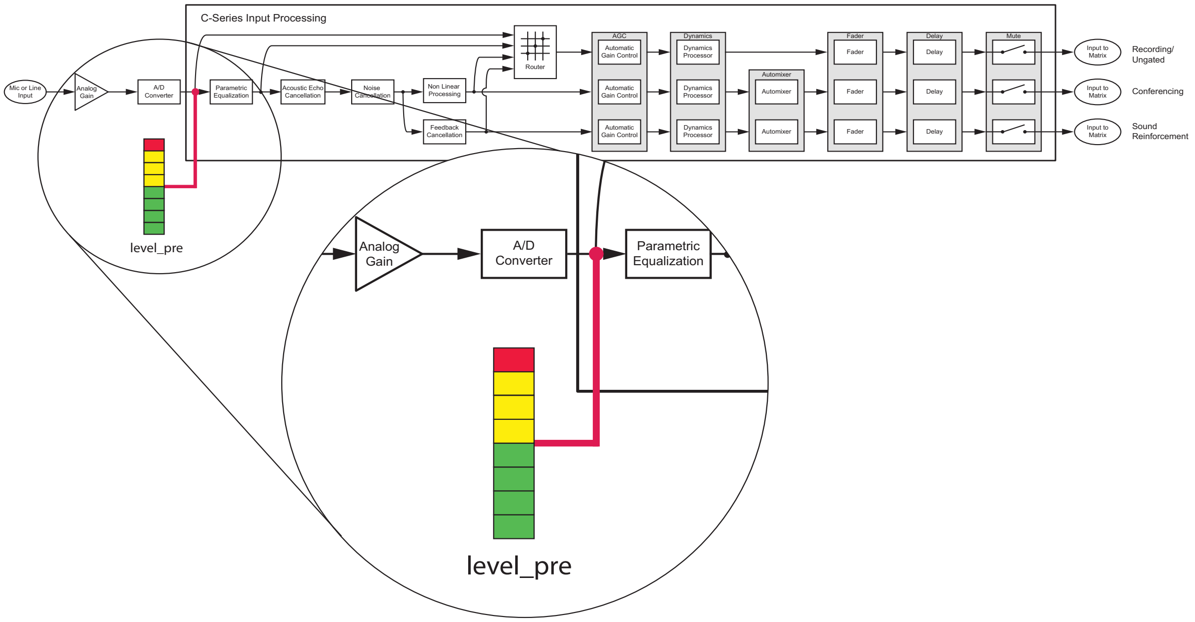
Understanding Meter Types

There are typically two types for meters that are available for each input channel - a level that is before any processing known as a `level_pre` and a level that is after any input processing known as `level_post`.

The `level_pre` meter always displays the signal level just after the A/D converter. This meter shows the effect of the analog signal gain before any digital processing takes place, as shown in the following figure.

Installing SoundStructure Devices discusses how the analog gain should be set for best performance. The `level_pre` for all input signals is shown in the following figure.

Analog Gain Signal Before (level_pre) Digital Processing


C-Series Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
Router
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
level_pre
Analog Gain
A/D Converter
Parametric Equalization
level_pre

The `level_pre` signal meter is adjacent to the analog input gain slider in SoundStructure Studio, as shown in the following figure. Adjustments to the gain slider are reflected in the meter - add more gain and the meter displays more signal activity; lower the gain, and the meter displays less signal activity.

The level_pre Signal Meter in SoundStructure Studio

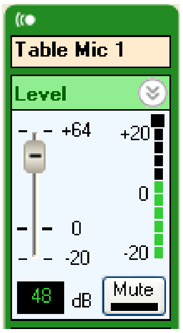


Table Mic 1
Level
+64
0
-20
+20
0
-20
48
dB
Mute

Because the level_pre meter position is before any processing is applied to the signal, even if the signal is muted within the SoundStructure device, the `level_pre` input meter displays any signal activity on that input.

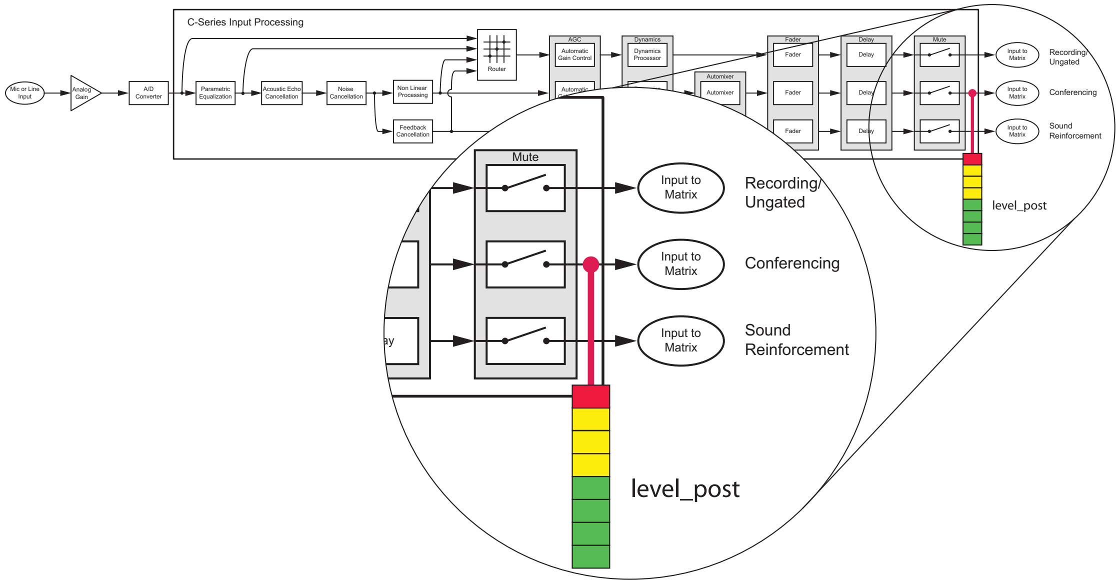
The `level_post` meter is after any processing, as shown in the following figure. In the example below, if the input signal is muted the `level_post` meter does not display any signal activity.

The exact location of the meter in the signal processing path depends on the type of signal that is viewed, as described next.

Measuring Microphone Post Levels

Microphone channels post level measure the signal level at the conferencing output of the input processing, as shown in the following figure.

Microphone Post Level Processing in SoundStructure Studio

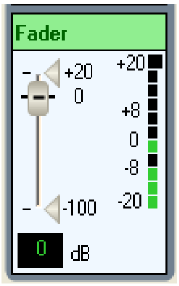


Microphone Post Processing Meter

C-Series Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
Router
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
level_post

You can use the fader on the bottom of the input channel to adjust the gain of the output of the input processing. The fader changes the level of all three outputs going to the matrix. The meter activity displays the affect of any gain adjustments.

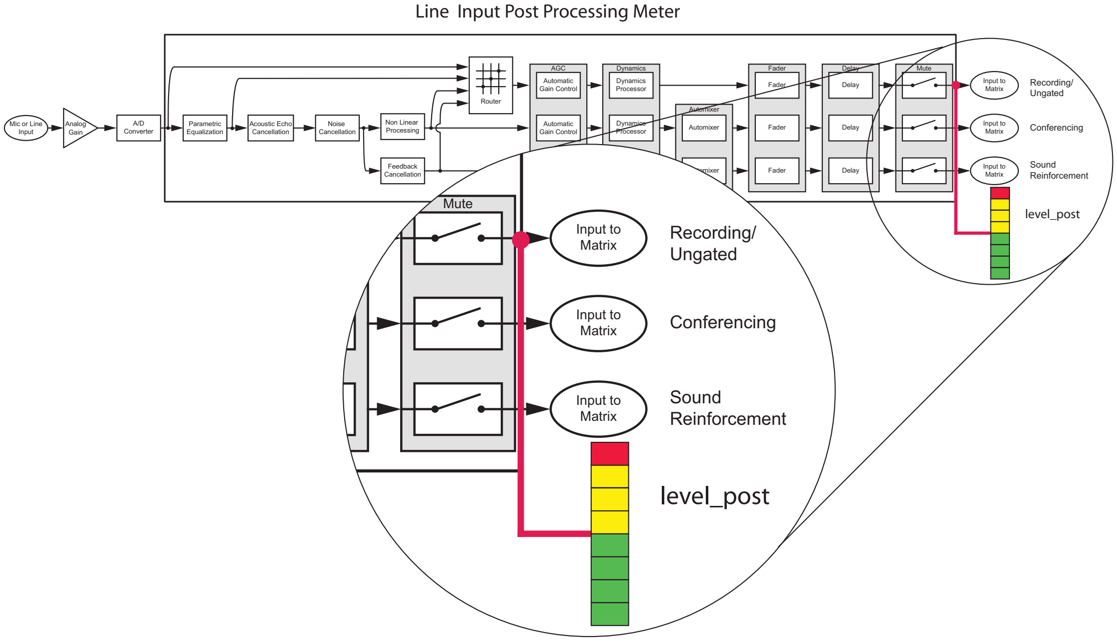
Input and Output Fader in SoundStructure Studio


Fader
+20
0
-100
+20
+8
0
-8
-20
0
dB

Metering Line Input Post Levels

Line input channels, such as program audio or audio from video codecs that are connected via analog inputs and outputs, are metered at the Recording/Ungated output, as shown in the following figure. Stereo virtual channels display two meters - one for each physical channel.

Line Input Channels Metered at the Recording/Ungated Output

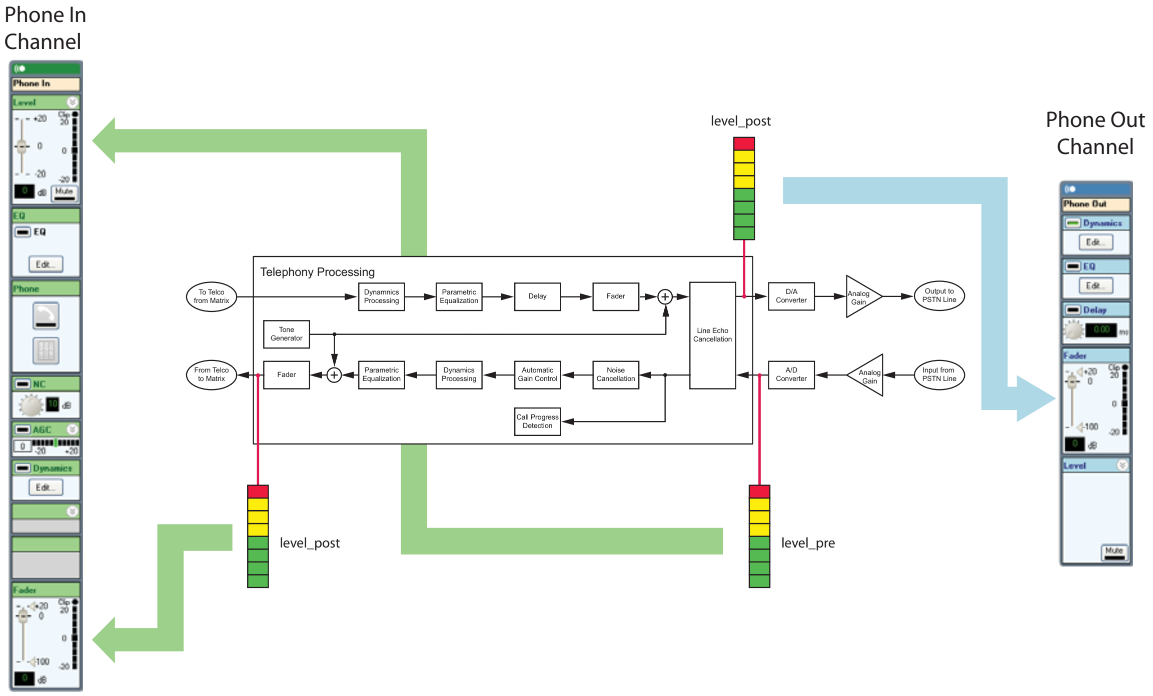

Line Input Post Processing Meter

Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
Router
AGC
Automatic Gain Control
Automatic Gain Control
Dynamics
Dynamics Processor
Dynamics Processor
Automixer
Automixer
Automixer
Fader
Fader
Fader
Fader
Delay
Delay
Delay
Delay
Mute
Input to Matrix
Recording/ Ungated
Input to Matrix
Conferencing
Input to Matrix
Sound Reinforcement
level_post
Mute
Input to Matrix
Recording/ Ungated
Input to Matrix
Conferencing
Input to Matrix
Sound Reinforcement
level_post

Processing with Telephony level_pre and level_post

For telephony channels, the `level_pre` and `level_post` for the phone input channel and level_post for the phone output channels are shown in the following figure. As with the analog input and output channels, the `level_pre` is before any processing and the `level_post` is after the processing.

level_pre and level_post Input and Output Processing for Telephony Channels

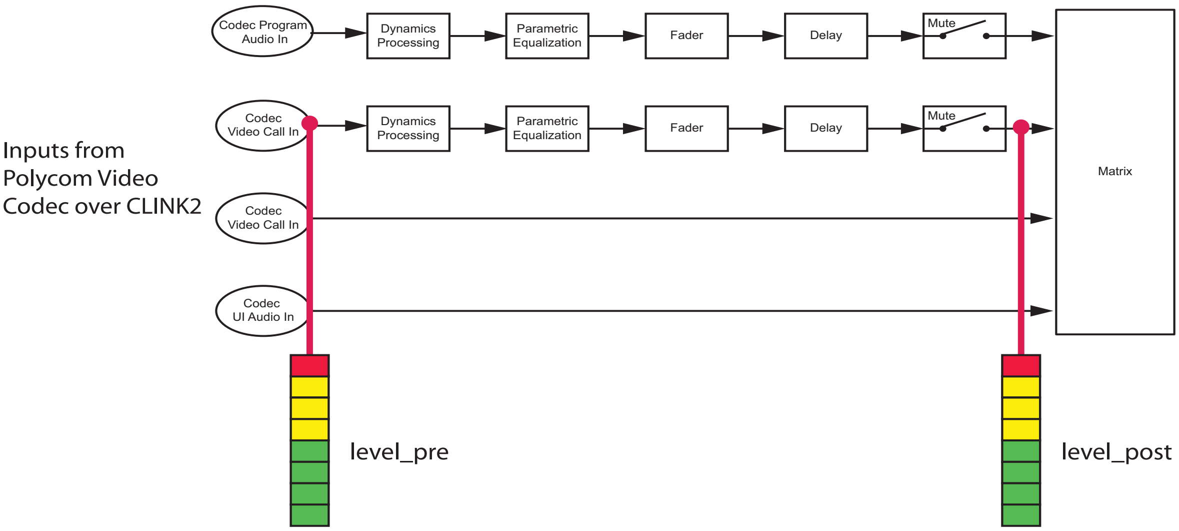

Phone In Channel
Phone In
Level
Mute
EQ
Edit...
Phone
NC
AGC
Dynamics
Fader
dB
level_post
Phone Out Channel
Phone Out
Dynamics
EQ
Delay
Fader
Level
Telephony Processing
To Telco from Matrix
Dynamnics Processing
Parametric Equalization
Delay
Fader
Tone Generator
Line Echo Cancellation
D/A Converter
Analog Gain
Output to PSTN Line
From Telco to Matrix
Fader
Parametric Equalization
Dynamics Processing
Automatic Gain Control
Noise Cancellation
A/D Converter
Analog Gain
Input from PSTN Line
Call Progress Detection
level_post
level_pre

Using Conference Link Channels

The Conference Link channels for Codec Program Audio in and Codec Video Call In have a `level_pre` and `level_post`, as shown on the following figure. The Codec Voice In and Codec UI Audio In channels do not have `level_pre` or `level_post` meters as those signals are available directly at the matrix and do not have any input processing on a SoundStructure device.

For more information on the processing available for the Conference Link2 channels, see Connecting Over Conference Link2.

level_pre and level_post Processing for Conference Link Channels


Inputs from Polycom Video Codec over CLINK2
Codec Program Audio In
Codec Video Call In
Codec Video Call In
Codec UI Audio In
Dynamics Processing
Parametric Equalization
Fader
Delay
Mute
Dynamics Processing
Parametric Equalization
Fader
Delay
Mute
Matrix
level_pre
level_post

Using Input Channel Controls

This section discusses the input controls in the order the channels display on the Channels page. The input channel settings are shown in the following figure in both a collapsed view and with the different areas expanded to show the additional controls.

You can also set any setting for a virtual channel can by adjusting the setting on a virtual channel group. By using virtual channel groups, the system can be setup very quickly because the parameters propagate to all the underlying virtual channels.

The input channel controls are expanded to show less frequently used controls such as phantom power, trim, delay compensation, and the selection of the different ungated signal types. See Introducing the Polycom SoundStructure Product Familyfor more information about the ungated/recording signal types and the signal processing that is available on those signal paths. More frequently used controls such as input gain and input fader are always available and are visible even when the control is collapsed.

Input Channel Settings in SoundStructure Studio

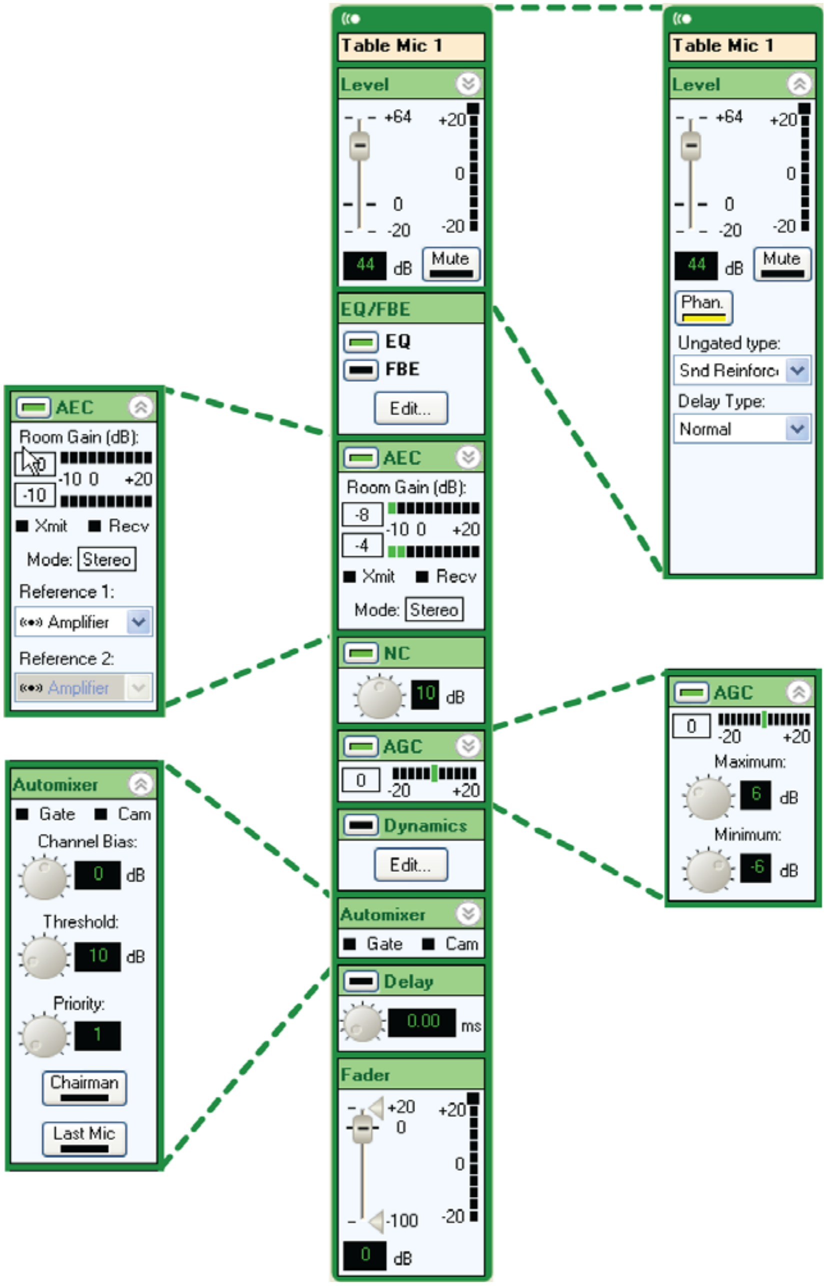


Table Mic 1
Level
+64
+20
0
-20
44 dB
Mute
EQ/FBE
EQ
FBE
Edit...
AEC
Room Gain (dB):
-8
-4
-10 0 +20
Xmit
Recv
Mode: Stereo
NC
10 dB
AGC
0
-20 +20
Dynamics
Edit...
Automixer
Gate
Cam
Delay
0.00 ms
Fader
+20
0
-100
0 dB
Table Mic 1
Level
44 dB
Mute
Phan.
Ungated type:
Snd Reinforce
Delay Type:
Normal
AEC
Room Gain (dB):
-10
-10 0 +20
Xmit
Recv
Mode: Stereo
Reference 1:
Amplifier
Reference 2:
Amplifier
AGC
0
-20 +20
Maximum:
6 dB
Minimum:
-6 dB
Automixer
Gate
Cam
Channel Bias:
0 dB
Threshold:
10 dB
Priority:
1
Chairman
Last Mic

Operating Analog Signal Gain

SoundStructure devices have a continuous analog input gain stage that operates on the analog input signal and has a range of -20 dB to +64 dB with 0.5 dB gain increments. Values are rounded to the nearest 0.5 dB. This continuous gain range is different from the gain Vortex products uses because the Vortex microphone inputs have a mic/line switch that adds 33 dB of gain to a Vortex input signal. As a result, 48 dB of gain on a SoundStructure input is equivalent to a gain of 15 dB on a Vortex mic/line input that is in mic mode because of the additional 33 dB of gain on the Vortex when in mic mode.

Since there is only one large input range on SoundStructure devices, it is easier to see how much gain is required for each microphone input.

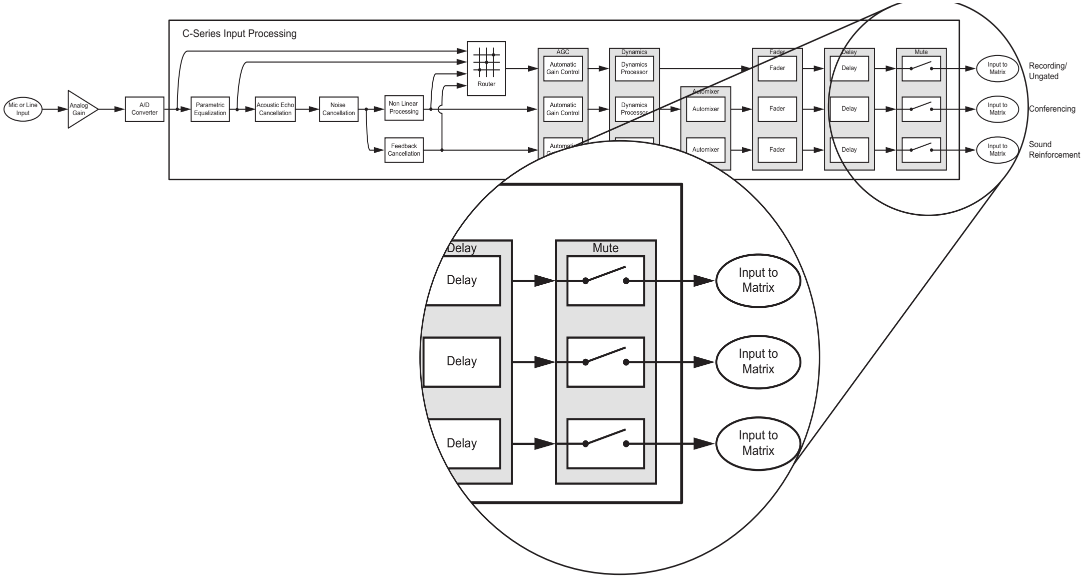
Gain settings are adjusted by moving the slider or typing the input value into the user control. Values can also be adjusted by clicking on the slider and using the up and down arrows to increase or decrease the value by 1 dB and by using the page up and page down keys to increase or decrease the value by 10 dB.

By supporting -20 dB as part of the analog gain range, effectively there is a 20 dB adjustable pad that makes it possible to reduce the gain of input sources that have a nominal output level that is greater than the 0 dBu nominal level of the SoundStructure devices.

Changing the Mute Status

You can change the mute status of an input virtual channel, or virtual channel group, by clicking **Mute**. When muted, the channel is muted after the input processing and before the input is used in the matrix, as shown in the following figure. The location of the input signal mute in the signal processing path ensures that the acoustic echo canceller, automatic gain control, feedback reduction, and noise canceller continue to adapt even while the input is muted.

Muted Channels Before and After Input Processing

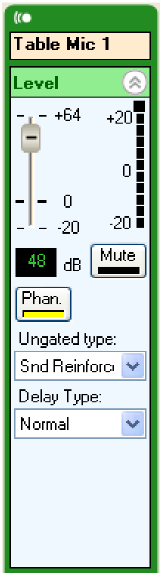

C-Series Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
Router
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement

Enabling Phantom Power

Enabled or disabled 48 V phantom power on a per input basis by clicking the phantom power button. The SoundStructure device supports up to 7.5 mA of current at 48 V on every input. By default, phantom power is turned off for all inputs if there is no SoundStructure Studio configuration loaded into the device.

To enable or disable the phantom power:

» Expand the level control by clicking on the expand graphic in the upper right corner and click the **Phan,** the phantom power button.



Ungated type: Snd Reinforc

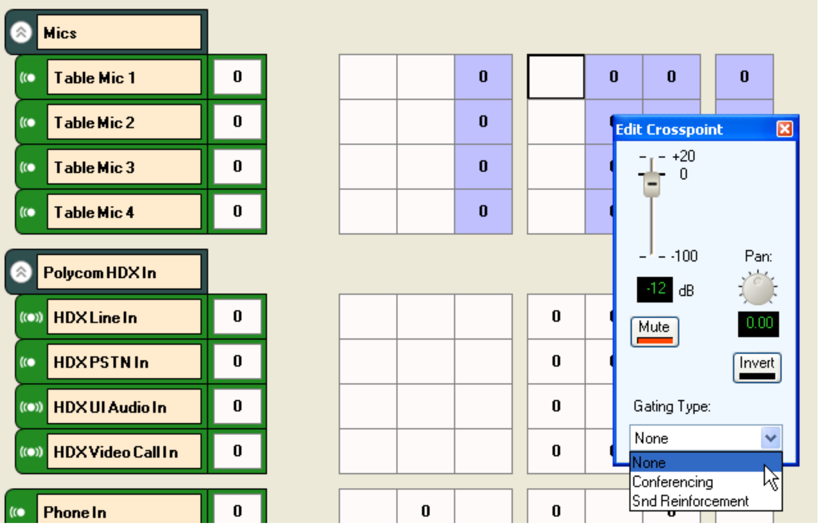
Delay Type: Normal

Using the Ungated Type

The ungated type user control refers to which signal path to use for the ungated (or un-automixed) processing path. The decision of whether to use the ungated version of the input channel processing is made at the matrix crosspoint, as shown in the following figure, where the gated type None is highlighted.

After the ungated signal is selected in the matrix, the decision of which ungated *type* of the signal is used is made on the channels page on an input by input basis.

Ungated Version of Input Channel Processing Matrix Crosspoint



Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Polycom HDX In
HDX Line In
HDX PSTN In
HDX UI Audio In
HDX Video Call In
Phone In
Edit Crosspoint
+20
0
-100
Mute
Invert
None
Conferencing
Snd Reinforcement

-12 dB

Pan: 0.00

Gating Type: None

As described in Introducing the Polycom SoundStructure Product Family, there are four types of ungated signal processing paths that can be selected for each input. The different signal processing paths for the four ungated signal types are summarized in the following table.

Summary of Ungated Signal Types

Ungated Type	Summary
Bypass	No signal processing on the audio channel.
Line Input	Equalization, dynamics processing, AGC
Conferencing	Equalization, echo and noise cancellation, non linear processing, dynamics processing, AGC
Sound Reinforcement	Equalization, echo and noise cancellation, feedback elimination, dynamics processing, AGC

The default ungated type depends on the type of input signal, as shown in the following table.

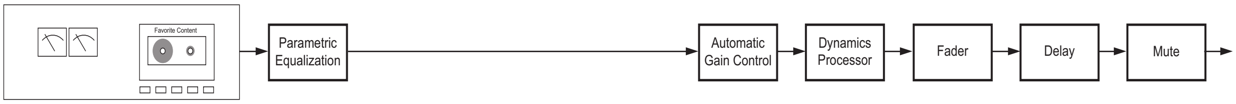
Signal Type and Default Ungated Type

Signal Type	Default ungated type
Microphone channels	Sound Reinforcement
Non microphone channels	Line input

Most applications benefit from the Line Input ungated signal processing path for program audio and other non-microphone audio that is not usually automixed.

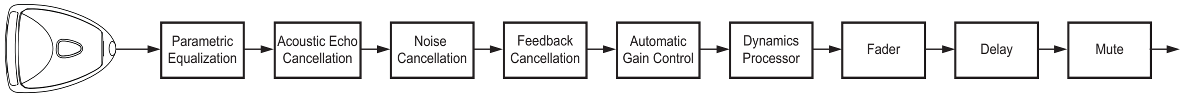
An example of using the line input processing is shown in the following figure where a program audio source can be processed with parametric equalization, automatic gain control, dynamics processing, fader, delay, and input mute.

Ungated Line Input Processing Example


Favorite Content
Parametric Equalization
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

The Sound Reinforcement path is selected by default for microphone audio because that processing path includes the full echo and noise cancellation, but the path does not include the non-linear processing associated with the acoustic echo canceller to avoid the application of any echo canceller suppression (or ducking) to the signal. The application of using this path is shown in the following figure where a microphone is connected and echo canceled and feedback reduced, but not automixed

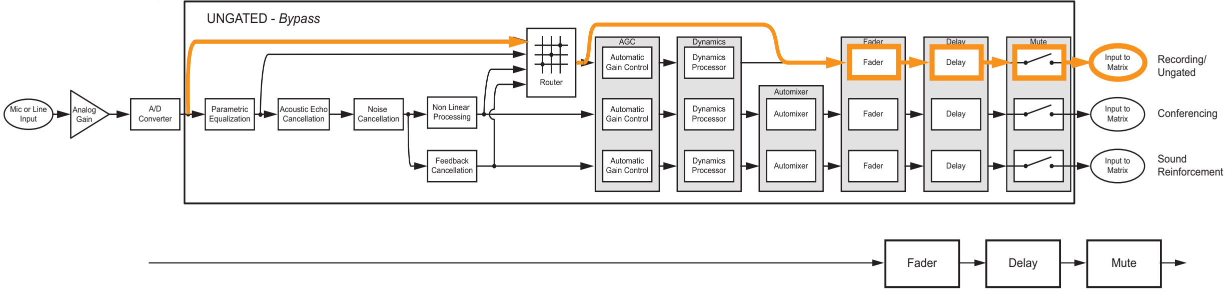
.Ungated Sound Reinforcement Processing Application


Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Feedback Cancellation
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

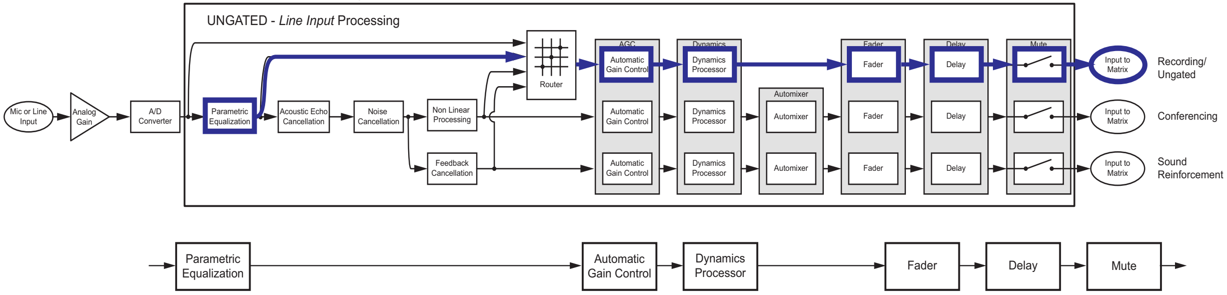
A complete summary of the signal processing associated with each ungated processing type is shown in the following figure. For additional information, see Introducing the Polycom SoundStructure Product Family.

Summary of Ungated Signal Processing

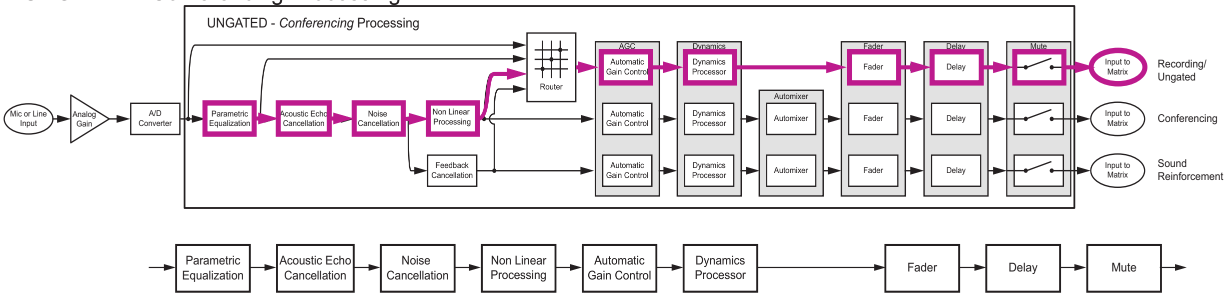
UNGATED - Bypass


UNGATED - *Bypass*
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
Router
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Fader
Delay
Mute

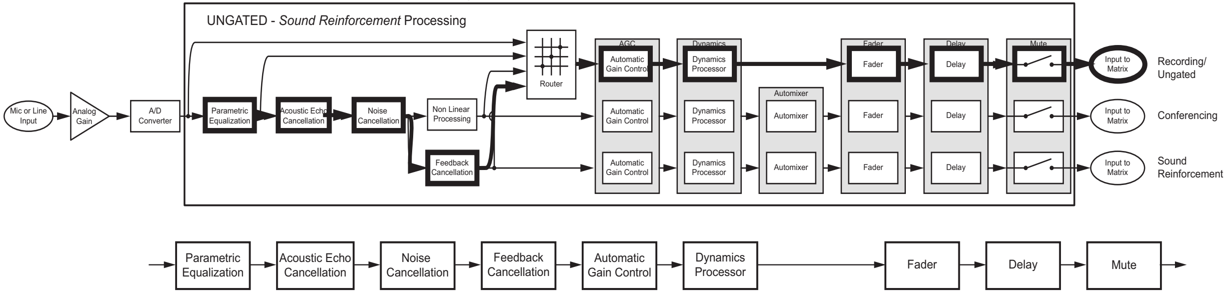
UNGATED - Line Input Processing


UNGATED - *Line Input* Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
Router
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Parametric Equalization
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

UNGATED - Conferencing Processing


UNGATED - *Conferencing* Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
Router
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

UNGATED - Sound Reinforcement Processing

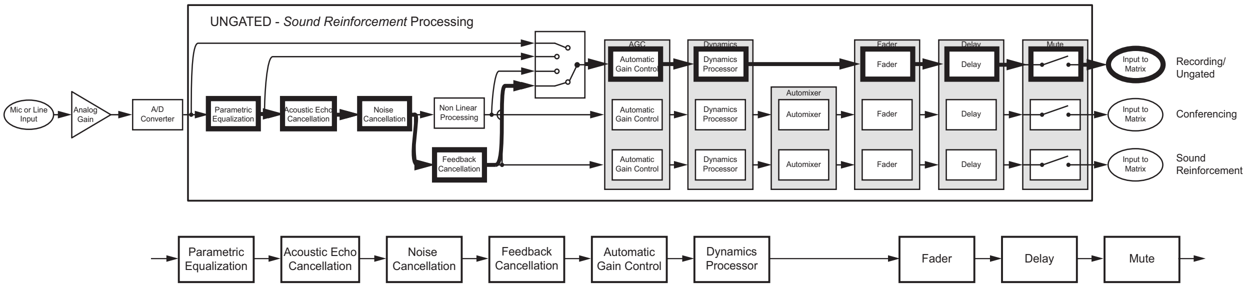

UNGATED - *Sound Reinforcement* Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
Router
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Feedback Cancellation
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

Using Delay Type

When you select the Sound Reinforcement ungated type, there are two delay options that are available on the Sound Reinforcement signal path: normal and low delay.

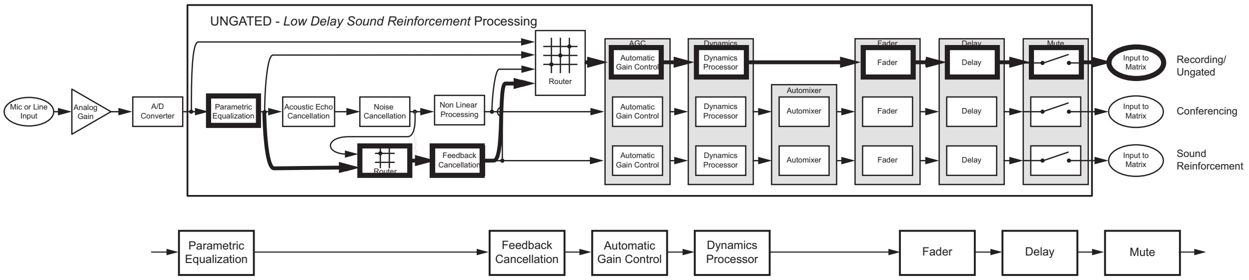
The normal delay type for the Sound Reinforcement ungated type corresponds to the processing paths that was defined previously and is shown in the following figure.

Sound Reinforcement Ungated Normal Delay Type Processing


UNGATED - *Sound Reinforcement* Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Feedback Cancellation
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

The *low delay* type corresponds to a processing path that completely bypasses the processing of the AEC and noise cancellation. Because these processing blocks are not in the signal path, the signal has lower latency. The AEC and noise cancellation add 16 MSEC of latency to the signal path. The resulting processing path from bypassing the AEC and noise cancellation paths is shown in the following figure.

Low Delay Type Processing Bypassing AEC and Noise Cancellation


UNGATED - *Low Delay Sound Reinforcement* Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Router
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Parametric Equalization
Feedback Cancellation
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

Note: No Echo and Noise Cancellation when Low Delay is Selected

When the low delay option is selected, the sound reinforcement and sound reinforcement ungated processing paths do not have any echo and noise cancellation processing. Only the conferencing and ungated conferencing versions of the input processing have echo and noise cancellation processing.

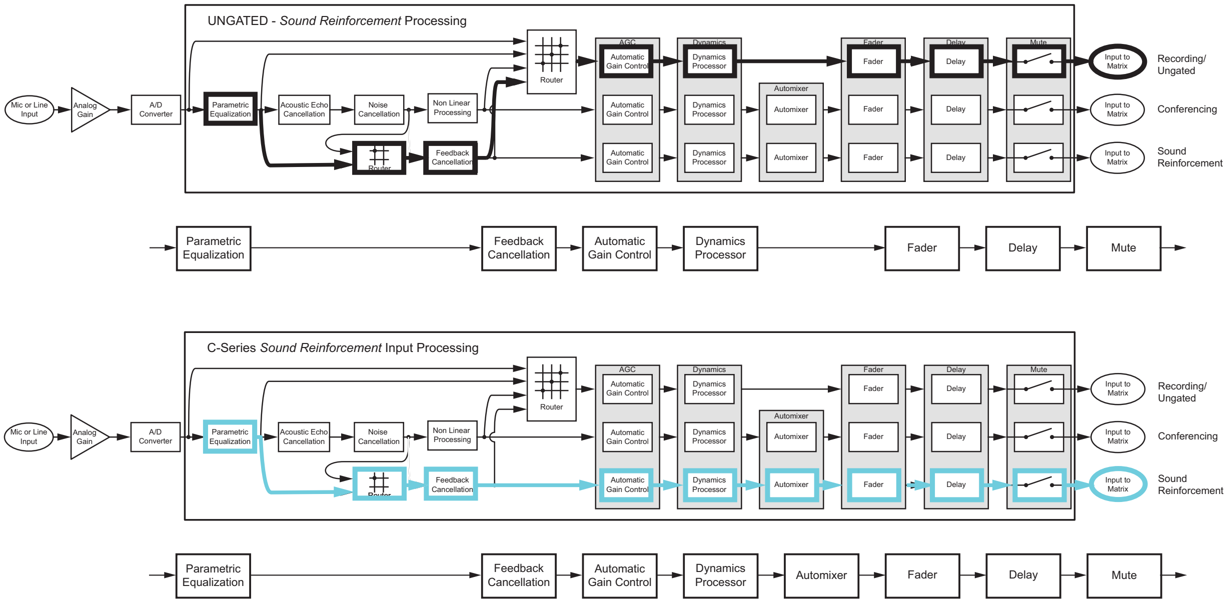
These two delay options are summarized in the following table.

Normal and Low Delay Type Options

Delay Type	Meaning
Normal	The signal path includes the latency associated with the echo and noise cancellation signal path
Low delay	The signal path does NOT include the latency associated with the echo and noise cancellation signal path. The echo and noise cancellation blocks are completely bypassed.

The signal processing associated with the low delay option is shown in the following figure for both the ungated sound reinforcement path and automixed sound reinforcement paths.

Signal Processing for the Low Delay for Ungated and Automixed Sound Reinforcement Paths


UNGATED - Sound Reinforcement Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Router
Feedback Cancellation
AGC
Automatic Gain Control
Dynamics
Dynamics Processor
Automixer
Fader
Delay
Mute
Input to Matrix
Recording/ Ungated
Conferencing
Sound Reinforcement
Parametric Equalization
Feedback Cancellation
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute
C-Series Sound Reinforcement Input Processing
Parametric Equalization
Feedback Cancellation
Automatic Gain Control
Dynamics Processor
Automixer
Fader
Delay
Mute

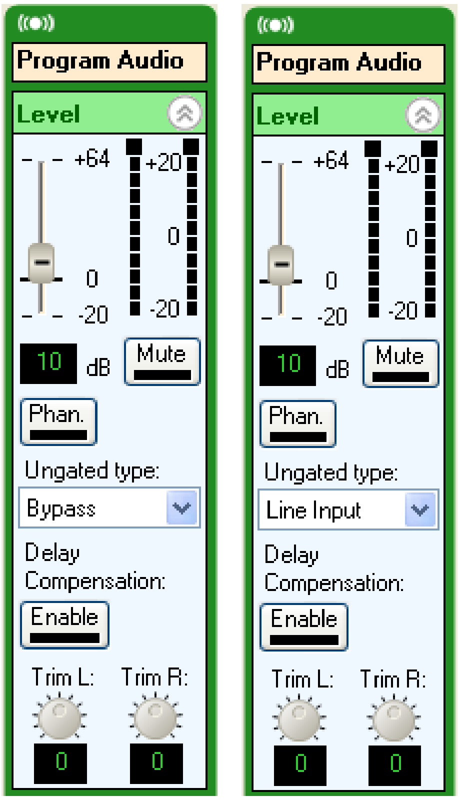
Using Delay Compensation

The delay compensation control adds a fixed delay to the line input and bypass signal processing paths to keep the different version of the input processing time aligned through the input processing.

Microphone inputs have approximately 16 msec of latency due to the AEC and noise cancellation processing. By selecting delay compensation, 16 msec of delay is added to the line input and bypass ungated signal types.

The option for the delay compensation displays when the Line Input or Bypass ungated signal type is selected, as shown in the following figure.

Line Input or Bypass Ungated Signal Type Delay Compensation Option


Program Audio
Level
+64
0
-20
+20
0
-20
10
dB
Mute
Phan.
Delay Compensation:
Enable
Trim L:
Trim R:
0
0
Program Audio
Level
+64
0
-20
+20
0
-20
10
dB
Mute
Phan.
Delay Compensation:
Enable
Trim L:
Trim R:
0
0

Ungated type: Bypass

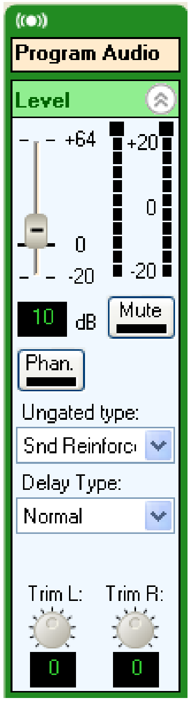
Ungated type: Line Input

Using Trim

The trim command is used with stereo virtual channels to provide additional gain or attenuation in the analog domain to the underlying left or right physical channels in case the incoming signal levels need to be adjusted separately. As shown in the following figure, there are two trim knobs for stereo virtual channels and no trim knob for mono virtual channels.

The trim gain applies in the analog input gain as long as the trim plus the analog input gain do not exceed 64 dB. Additional trim gain beyond a total gain of 64 dB is added in the digital domain.

Trim Knobs for Virtual Channels



10 dB

Ungated type: Snd Reinforc

Delay Type: Normal

Trim L: 0

Trim R: 0

Processing Equalization

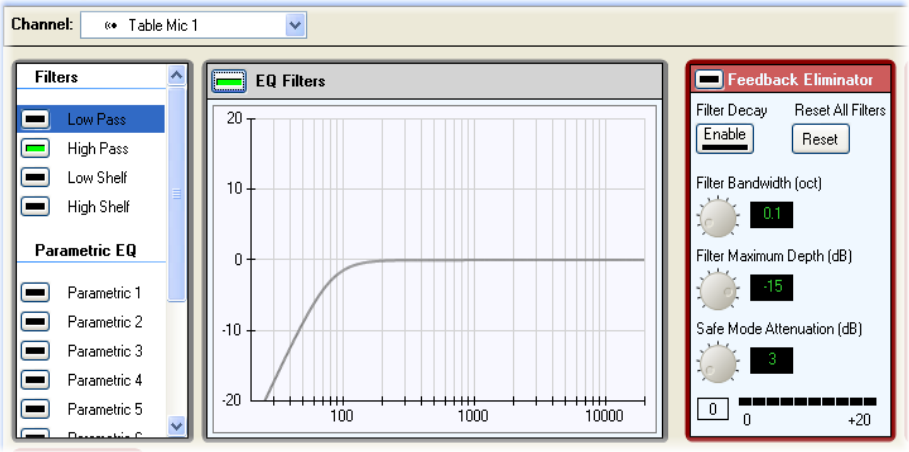
The equalization processing that is available for each input consists of the following dedicated filters and equalizers:

- Low Pass filter
- High Pass filter
- Low Shelf filter
- High Shelf filter
- 10 parametric equalizers.

These filter types are shown in the following figure. The overall equalization processing are enabled or disabled using the button next to the EQ block name on the Channels page or equivalently by using the button next to the EQ Filters text, as shown in the following figure.

The equalization page also displays the feedback elimination user controls and a list of frequencies where feedback is found when the processing is enabled.

Dedicated Filters and Equalizers for Equalization Processing



Channel: Table Mic 1

Filter Bandwidth (oct): 0.1

Filter Maximum Depth (dB): -15

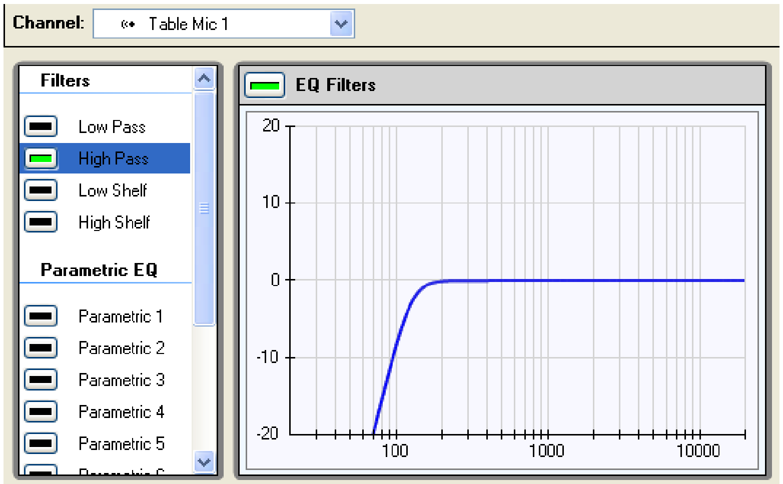
Safe Mode Attenuation (dB): 3

Filters
Low Pass
High Pass
Low Shelf
High Shelf
Parametric EQ
Parametric 1
Parametric 2
Parametric 3
Parametric 4
Parametric 5
EQ Filters
20
10
0
-10
-20
100
1000
10000
Feedback Eliminator
Filter Decay
Enable
Reset All Filters
Reset
0
0
+20

To enable a filter, click ▬ next to a filter, and adjust the parameters for the filter block, as shown in the following figure.

You can adjust the cut off frequency of the Low Pass and High Pass filters to between 0 Hz and 20,000 Hz, adjust the order from 2nd to 8th, and either select a Butterworth or Linkwitz-Riley filter .

Editing a High Pass Equalization Filter



Channel: Table Mic 1

Filters
Low Pass
High Pass
Low Shelf
High Shelf
Parametric EQ
Parametric 1
Parametric 2
Parametric 3
Parametric 4
Parametric 5
EQ Filters
20
10
0
-10
-20
100
1000
10000

For the parametric EQ filters, you can choose from:

- Parametric filter
- Notch filter
- Allpass filter

Parametric filters emphasize or de-emphasize the center frequency with a gain and bandwidth setting. You can specify the bandwidth (in octaves), center frequency (in Hz), and gain (from 0 to 20 dB).

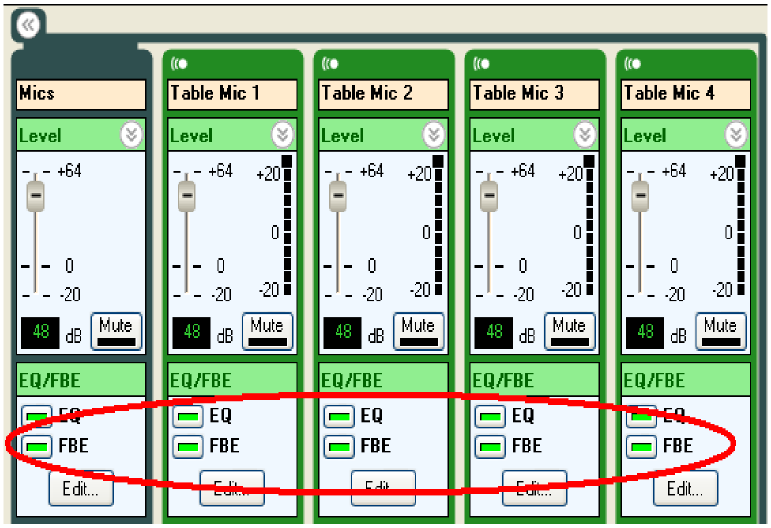
Notch filters eliminate energy (attenuate only) at the center frequency. The amount of attenuation for the signal is determined by the bandwidth (in octaves) selected. The bandwidth is defined as where the gain is -3 dB.

Allpass filters do not modify the gain of the signal, but change the phase. For a second order allpass filter, the phase shift is 0 degrees at 0 Hz, 360 degrees at high frequencies, and 180 degrees at the center frequency. The bandwidth is defined as the bandwidth (in octaves) where the phase shift is 90 degrees and 270 degrees.

Eliminating Feedback

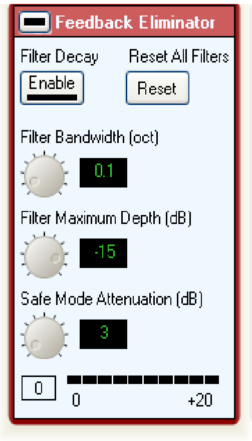
Feedback elimination uses 10 adaptive filters to reduce feedback that may be picked up by the microphone. When the feedback cancellation processing is enabled for a particular virtual channel, you can adjust the filter bandwidth from 0.03 to 1 octave and the filter depth from 0 to -100 dB. When enabled, the user interface displays the FBE as enabled, as shown in the following figure.

Feedback Elimination Enabled


Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Level
+64
+20
0
-20
48
dB
Mute
EQ/FBE
EQ
FBE
Edit...

Selecting **Edit** opens the equalization user control where parameters for the feedback eliminator are specified, as shown in the following figure.

Feedback Elimination Parameters in the Equalization User Control


Feedback Eliminator
Filter Decay
Enable
Reset All Filters
Reset
Filter Bandwidth (oct)
0.1
Filter Maximum Depth (dB)
-15
Safe Mode Attenuation (dB)
3
0
0
+20

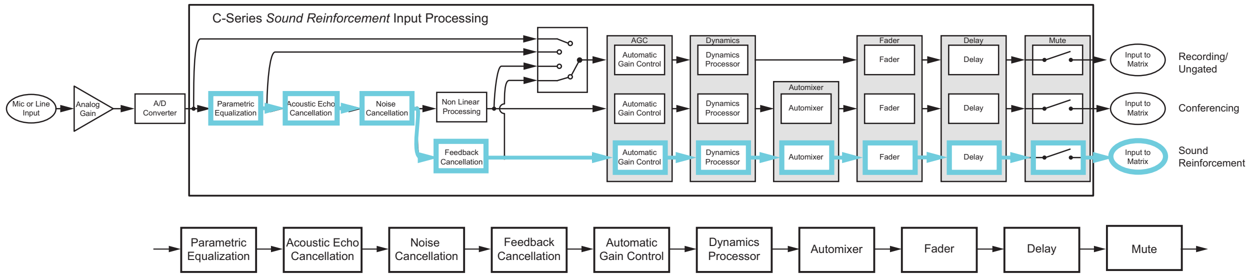
There is a safe mode attenuation that defines the amount of attenuation that are applied to the signal if the feedback eliminator filters are all engaged and there is still feedback. The safe mode attenuation can be set from 0 to 20 dB of attenuation and have a default value is 3 dB.

The Filter Decay control allows the adaptive filters to relax as the feedback is reduced in the system.

During operation, if persistent frequencies appear, you can fix the filter settings from those offending frequencies by clicking **Make Fixed**. This transfers the settings of the adaptive filter to one of the fixed parametric filters.

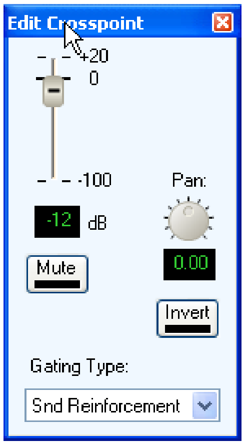
To utilize the feedback processing, you must enable the feedback processing on the EQ page for the desired inputs and select the sound reinforcement signal processing path Recall that the input processing has different types of audio processing available for the input signals. The sound reinforcement signal path for the C-series products is shown in the following figure.

C-Series Sound Reinforcement Signal Path


C-Series *Sound Reinforcement* Input Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
AGC
Automatic Gain Control
Automatic Gain Control
Automatic Gain Control
Dynamics
Dynamics Processor
Dynamics Processor
Dynamics Processor
Automixer
Automixer
Automixer
Fader
Fader
Fader
Fader
Delay
Delay
Delay
Delay
Mute
Input to Matrix
Recording/ Ungated
Input to Matrix
Conferencing
Input to Matrix
Sound Reinforcement
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Feedback Cancellation
Automatic Gain Control
Dynamics Processor
Automixer
Fader
Delay
Mute

You can select the sound reinforcement signal path at the matrix crosspoint by selecting the **Snd Reinforcement** option of the gated/automixed, as shown in the following figure. Selecting the Snd Reinforcement option ensures that the proper input processing path is selected for routing microphones to loudspeakers.

Snd Reinforcement Option of Gated/Automixed Sound Reinforcement Signal Path


Edit Crosspoint
+20
0
-100
Pan:
Mute
Invert

-12 dB

0.00

Gating Type: Snd Reinforcement

Note: Using the Feedback Processing

To use the feedback processing, enable the processing from the EQ page and select the sound reinforcement version of input processing path in the matrix.

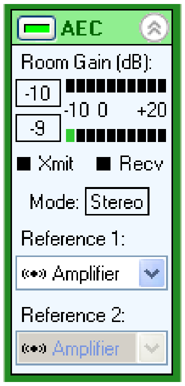
Enabling Acoustic Echo Cancellation (AEC)

The AEC is enabled/disabled by toggling the AEC button. The AEC control displays the mode of the acoustic echo canceller with Xmit indicating the system is transmitting audio to the remote site and Recv indicating the system is receiving audio from the remote site and is heard in the local room.

The room gain is shown graphically in the meter and the number in the box next to the meter. Room gain is defined in more detail in Appendix B: Address Book. The display of room gain is limited from -10 dB to +20 dB. See Installing SoundStructure Devices for additional information on room gain.

The AEC references for each input are specified in the pull-down combination boxes for the associated input signal. As described in the Creating Virtual Channel Groups section, you can select the AEC reference for the entire virtual channel group and that information propagates to all the virtual channels of the group.

Input Signal AEC References



Mode: Stereo

Reference 1: Amplifier

Reference 2: Amplifier

You can select references from any output signal or from any submix signal. A reference can either be a mono virtual channel or a stereo virtual channel. If you specify only a single mono virtual channel reference, the system operates as a monaural echo canceller. If you specify either a stereo virtual channel or two mono virtual channels, the system operates as a stereo echo canceller.

References need to consist of all the remote audio that is being played into the local room including telephony signals, video codec signals, and program audio.

Note: Using Output and Submix Signals as Echo Canceller References
You can use any output signal or submix signal as an echo canceller reference. The reference needs to include all remote audio sources.

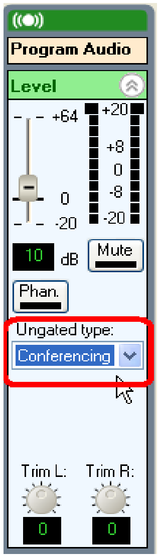
Processing Noise Cancellation

The noise cancellation processing is available on all analog inputs via the conferencing and sound reinforcement processing paths. The noise cancellation reduces background noise that is picked up by microphones or already present in input signals from program audio sources. You can turn the noise cancellation functionality on or off with the enable button and adjust the amount of noise cancellation c from 0 to 20 dB.

The SoundStructure noise cancellation effectively removes different types of background noise ranging from narrow band noise (e.g., tones) to broadband noise. For best performance, the noise characteristics need to be quasi-stationary. For example, the statistics of the underlying noise are fixed or change slowly over time.

You can enable noise cancellation for a non-microphone channel, such as a video codec audio or program audio, by selecting the conferencing version of the ungated signal path. Note that the default selection for non-microphone audio sources is the line-input processing path.

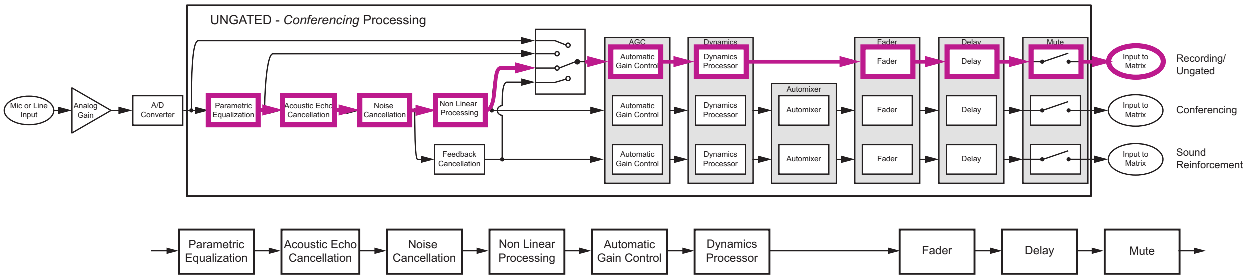
Conferencing as the Ungated Signal Path for Program Audio


Program Audio
Level
+64
0
-20
+20
+8
0
-8
-20
10
dB
Mute
Phan.
Trim L:
Trim R:
0
0

Ungated type: Conferencing

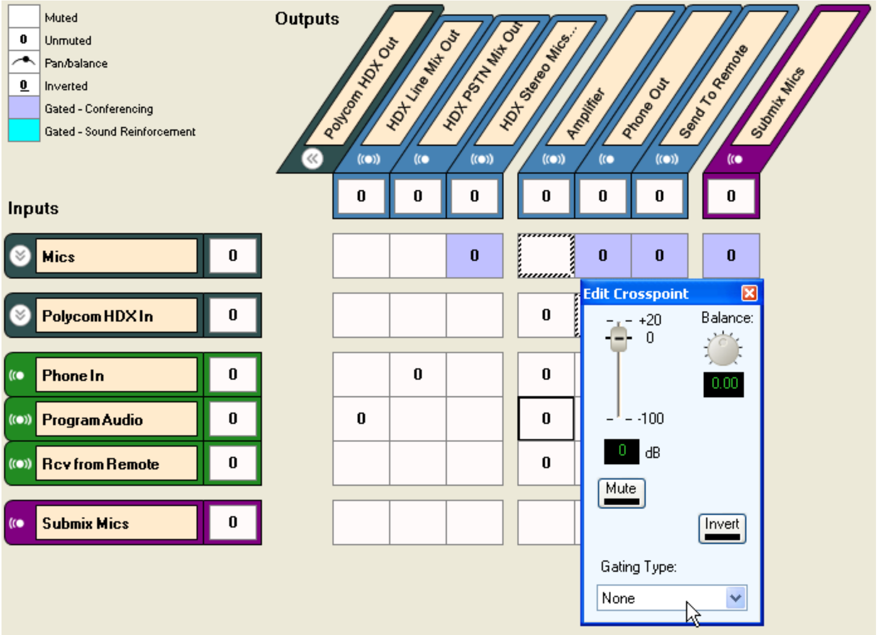
The ungated conferencing signal path is shown in the following figure. Notice that the noise cancellation processing is now in the signal path along with the automatic gain control, dynamics processing, fader, delay, and mute. The acoustic echo canceller is also in this signal path but is not enabled for non-microphone audio sources.

Ungated Conferencing Signal Path


UNGATED - *Conferencing* Processing
Mic or Line Input
Analog Gain
A/D Converter
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Feedback Cancellation
AGC
Automatic Gain Control
Automatic Gain Control
Automatic Gain Control
Dynamics
Dynamics Processor
Dynamics Processor
Dynamics Processor
Automixer
Automixer
Automixer
Fader
Fader
Fader
Fader
Delay
Delay
Delay
Delay
Mute
Input to Matrix
Recording/ Ungated
Input to Matrix
Conferencing
Input to Matrix
Sound Reinforcement
Parametric Equalization
Acoustic Echo Cancellation
Noise Cancellation
Non Linear Processing
Automatic Gain Control
Dynamics Processor
Fader
Delay
Mute

After you select the conferencing ungated type in the Channels page, select the ungated signal path in the matrix, as shown in the following figure. This selection chooses the conferencing ungated signal path and enables you to enable noise cancellation on that input signal.

Ungated Signal Path in Matrix


Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Polycom HDX Out
HDX Line Mix Out
HDX PSTN Mix Out
HDX Stereo Mics...
Amplifier
Phone Out
Send To Remote
Submix Mics
Inputs
Mics
Polycom HDX In
Phone In
Program Audio
Rcv from Remote
Submix Mics
Edit Crosspoint
+20
0
-100
Balance:
0.00
0 dB
Mute
Invert
Gating Type:
None

Using Automatic Gain Control (AGC)

Automatic gain control is used to automatically adjust the gain of audio signals so that the average signal level is close to the SoundStructure nominal signal level of 0 dBu. You can use the AGC processing on any input signal.

AGC is typically used on microphone input signals to compensate for local talkers that are different distances from their microphones or telephone input signals to compensate for varying telephone levels. The AGC system is designed to adapt the gain only when valid speech is present.

You can turn the AGC on or off with the AGC enable button. When the AGC is enabled, you can view the current AGC gain (or attenuation) from the AGC meter and the text box next to the AGC meter.

You can adjust the range of the AGC by expanding the AGC control and adjusting the maximum and minimum gains. By default, the maximum and minimum gain are set to 6 and -6 respectively on microphone and telephony signals.

The maximum AGC value specifies the maximum amount of gain the AGC can apply to increase the input signal level as the AGC tries to reach the SoundStructure nominal signal level.

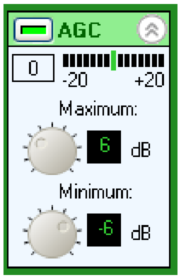
The minimum AGC value specifies the maximum amount of attenuation the AGC can apply to attenuate the input signal as the AGC tries to reach the SoundStructure nominal signal level.

If the input is a stereo virtual channel, the AGC gain for both underlying left and right physical channels uses the same gain, ensuring that the stereo image is preserved.

To operate the AGC with a target level different from 0 dBu:

1. Set the AGC minimum and maximum range to the desired range
2. Adjust the input fader to the desired target level above or below the 0 dBu nominal signal level of the SoundStructure devices.

   This allows the AGC to adapt to the 0 dBu nominal level and the fader settings offset the 0 dBu level to the setting on the fader.



Using Dynamics Processors

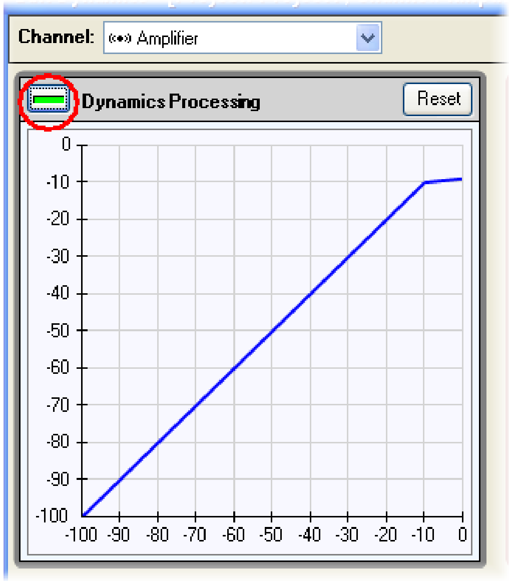
Dynamics processors, also known as non-linear processors, are used to reduce the dynamic range, or amplitude, of input or output signals and are often used on sound reinforcement systems to prevent clipping audio amplifiers. Dynamics processors are similar to automatic gain controllers, but are typically faster acting and can be used with program audio and other fast changing input signals.

SoundStructure devices include the following styles of look-ahead dynamics processing:

- Peak Limiter
- Limiter
- Compressor
- Expander
- Gate

The SoundStructure Studio user interface for adjusting the dynamics settings are shown in the following figure.

Dynamics Processing Adjustment

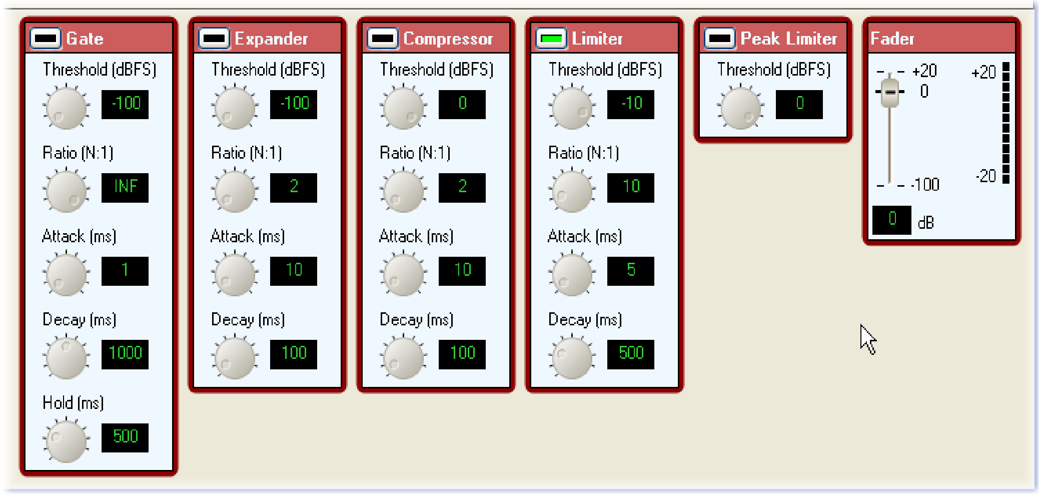


Channel: Amplifier

You can turn the dynamics processing turned on or off for a channel by toggling the enable button on either the channels page or on the dynamics control highlighted on the previous figure. As with other controls, you can configure dynamics processing for a virtual channel or a virtual channel group. When dynamics processing is applied to stereo virtual channels, the underlying left and right dynamics processors apply the same gain. Linked dynamics processors apply gain as though the highest level input signal were applied to both of their inputs.

You must enable the gate, expander, compressor, limiter, and peak limiter individually with their individual enable buttons. In the following figure the limiter is enabled.

Enabled Limiter for Dynamic Processing



Gate
Threshold (dBFS) -100
Ratio (N:1) INF
Attack (ms) 1
Decay (ms) 1000
Hold (ms) 500

Expander
Threshold (dBFS) -100
Ratio (N:1) 2
Attack (ms) 10
Decay (ms) 100

Compressor
Threshold (dBFS) 0
Ratio (N:1) 2
Attack (ms) 10
Decay (ms) 100

Limiter
Threshold (dBFS) -10
Ratio (N:1) 10
Attack (ms) 5
Decay (ms) 500

Peak Limiter
Threshold (dBFS) 0

Fader
0 dB

After the dynamics processing is enabled, the dynamics processing curve updates as adjustments are made to the dynamics processing settings.

On the dynamics processing page there is also a fader control - the same fader control found on the channels and matrix page- that can be used to add or remove gain from the underlying virtual channel.

The Reset button may be used to return the Dynamics processing to its default settings which leaves the signal unprocessed.

Using Compressors And Limiters

The peak limiter monitors the peak signal magnitude and compares it to a threshold. If the peak surpasses the threshold, the peak limiter immediately attenuates the signal with a very fast attack to bring the peak level below the threshold.

Limiters and compressors attenuate high-level signals without changing low level signals and are typically used to prevent loud signals from clipping, or to reduce the dynamic range of a signal to make the output level more consistent even if the input level is not consistent. When the input signal level rises above the compressor’s threshold, the compressor applies attenuation so that the output signal increases at a rate of one over the compression ratio past the threshold. Signals below the threshold are not modified, signals above the threshold are “compressed” or scaled by the compression ratio.

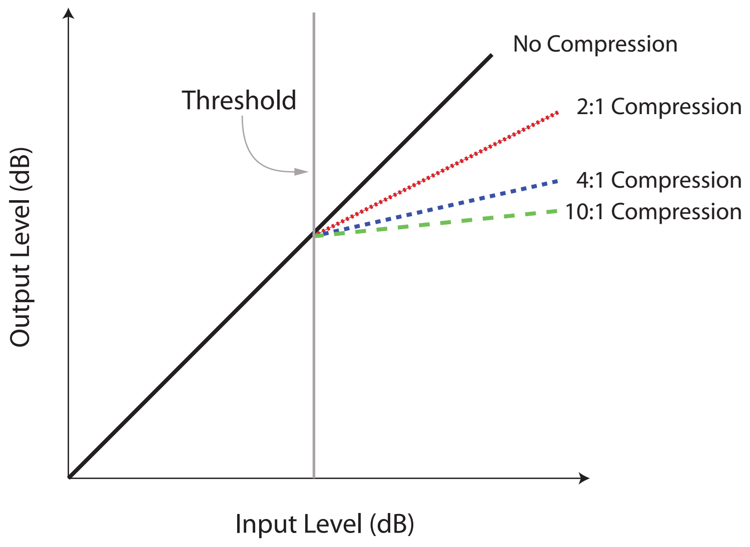
For example, if the compression ratio is set to 4:1, the threshold is set to -10 dBFS[1], and the input signal level is -2 dBFS (8 dB above the threshold) the compressor applies the compression ratio (in this case 4:1) and divides the 8 dB by 4 to arrive at 2 dB. The output signal is then -8 dBFS (2 dB above the threshold) even though the input signal was 8 dB above the threshold.

From this example, it is clear that the threshold is not a hard limit, but rather the onset of when the “compression” or division by the compression ratio is engaged. This is shown in the following figure.

---

1. dBFS means dB full scale where 0 dBFS is the maximum input signal allowed. Due to the SoundStructure design of a nominal signal level of 0 dBu with 20 dB headroom, -20 dBFS equals 0 dBu.

Engaged Compression Ratio



The “attack” portion of the compressor is when attenuation is increased as the signal level crosses the threshold, and the “decay” portion is when the attenuation is reduced toward 0 dB as the signal level falls below the threshold. Decreasing the attack time allows the compressor/limiter to work more aggressively but may also introduce audio artifacts.

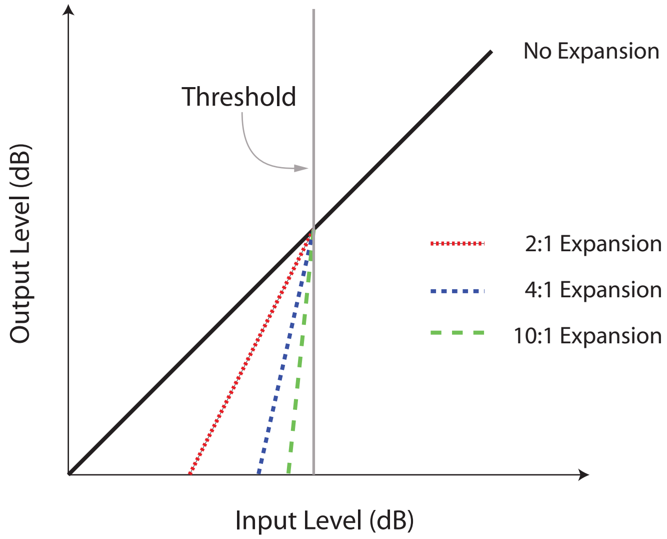
Limiters perform just like compressors, but are typically set with higher compression ratios (10:1 or more) to further limit the dynamic range of signals levels above the threshold.

Using Gates and Expanders

Expanders and gates are another form of dynamics processing that attenuate low level signals and leave the high level signals alone. This expands or increases the dynamic range of a signal. When the input signal level falls below the expander’s threshold, it applies an amount of attenuation (in dB) equal to the expansion ratio times the difference between the threshold and the signal level, as shown in the following figure.

For example, if the expansion ratio is 4:1, the threshold is -30 dBFS, and the input signal level is -35 dBFS, then the expander applies 20 dB of attenuation (4 x (35-30) = 20). When the signal is above the expander threshold, a gain of 1 is applied to the signal, therefore, the input signal is left unchanged.

Input Signal Attenuation, Expansion Ratio, and Signal Level



The “attack” portion of the expander is when the attenuation is reduced toward 0 dB, and the “decay” portion is when the attenuation is increased.

Gates perform like expanders, but are typically set with higher expansion (that is, gate) ratios and have a longer hold time. The gate does not decay until the signal is lower than the threshold for longer than the hold time. This prevents the gate from attenuating the signal between short pauses in speech.

The gate threshold is the RMS level in dBFS of the input signal below which the gain turns on. The level must be below this threshold longer than the gate hold time before the gain begins to apply a gain change.

The gate ratio is the multiplier applied to the difference between the current input signal level and the gate threshold. For example, if the gate ratio is 10:1 and the input signal level is 6 dB below the gate threshold, the gate applies 60 dB of attenuation.

The gate attack is the amount of time it takes the gate to ramp the gain to the target gain once the input signal level surpasses the gate threshold.

The gate decay controls how quickly the gain ramps down once the signal level is lower than the gate threshold and the gate hold time has expired.

The expander threshold is the RMS level in dBFS of the input signal that when below this threshold, the expander engages. The expander ratio is the multiplier applied to the difference between the current input signal level and the expander threshold. For example, if the expander ratio is 2:1 and the input signal level is 3 dB below the expander threshold, the gate applies -6 dB of gain (equivalently 6 dB of attenuation). If the input signal level is above the expander threshold, a gain of 1 (0 dB), is applied to the input signal.

The expander attack time is the amount of time (in milliseconds) it takes the expander to ramp the gain up to the target level once the input signal exceeds the expander threshold.

Using Automatic Microphone Mixing

SoundStructure devices can use either gain sharing or gating styles of automatic microphone mixers and support up to sixty-three different automixer groups. Microphones in the same group are part of the same automixer and affect each others' gain or gating behavior. Each microphone input can be in one automixer group. The default automixer style used is gain-sharing.

Defining Automixer Groups

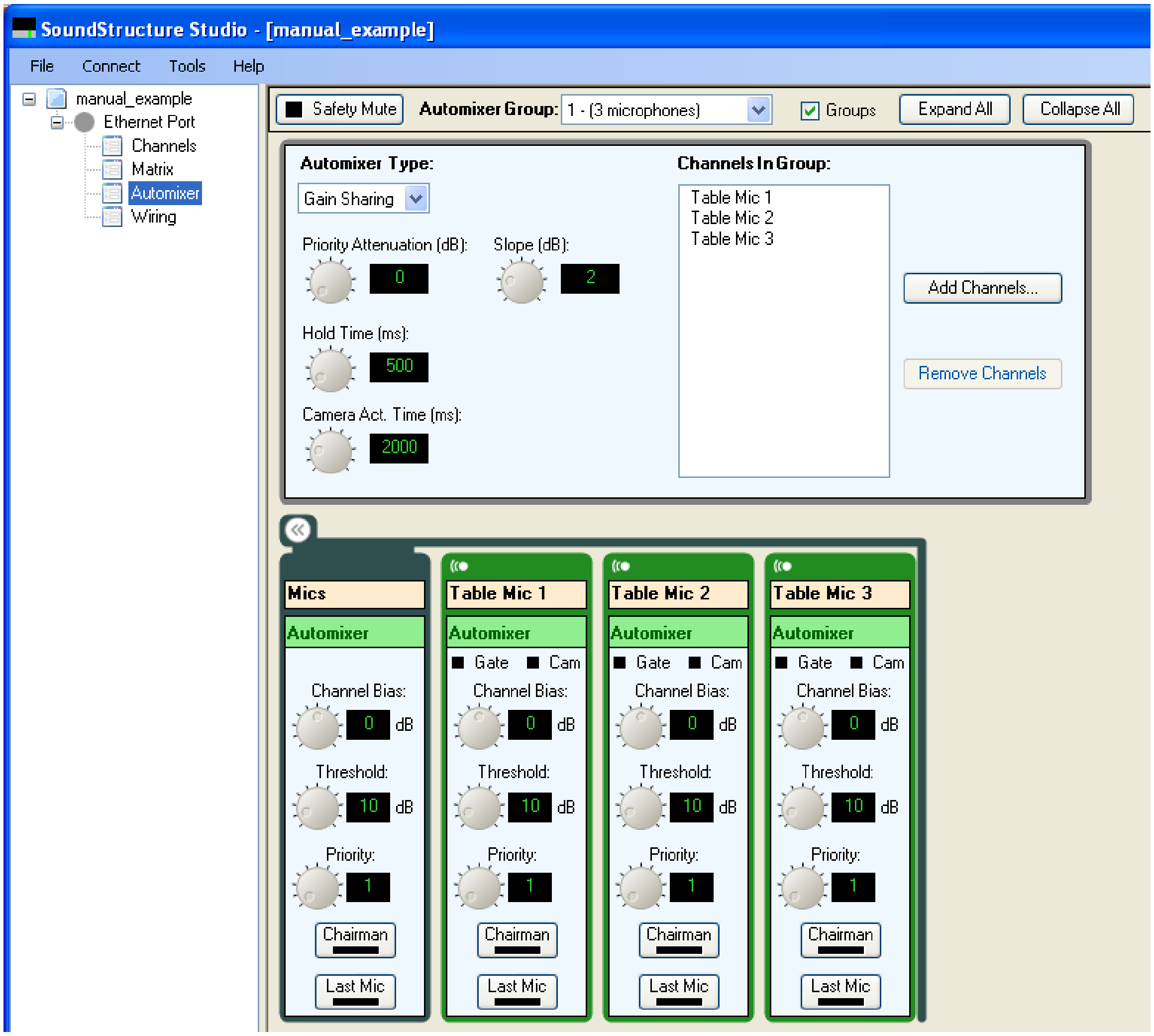
Generally, all of the microphones in one room should be in the same group, and microphones in different rooms should be in different groups. Even in zoned audio systems, all microphones should be in the same automixer group.

In room division applications, microphones in different rooms should be in different automixer groups when the rooms are divided. When the rooms are combined, they should be in the same automixer group. Changing the group for microphones can be easily done by creating virtual channel groups of the microphones in each room and when the rooms are combined, the **`am_group`** command can be used to set the new automixer group for the virtual channel group associated with the individual room's microphones. See Creating Advanced Applications for more information on room combining applications.

Using Automixer Controls

The SoundStructure Studio user controls for configuring the automixers are shown in the following figure. You can add channels to the automixer group by selecting **Add Channels**. You can remove channels by selecting **Remove Channels**.

Automixer Controls in SoundStructure Studio


SoundStructure Studio - [manual_example]

Automixer Group: 1 - (3 microphones)

☑ Groups

Automixer Type: Gain Sharing

Priority Attenuation (dB): 0

Slope (dB): 2

Hold Time (ms): 500

Camera Act. Time (ms): 2000

Channels In Group:

Table Mic 1
Table Mic 2
Table Mic 3

Mics	Table Mic 1	Table Mic 2	Table Mic 3
Automixer	Automixer	Automixer	Automixer
	■ Gate ■ Cam	■ Gate ■ Cam	■ Gate ■ Cam
Channel Bias: 0 dB	Channel Bias: 0 dB	Channel Bias: 0 dB	Channel Bias: 0 dB
Threshold: 10 dB	Threshold: 10 dB	Threshold: 10 dB	Threshold: 10 dB
Priority: 1	Priority: 1	Priority: 1	Priority: 1
Chairman	Chairman	Chairman	Chairman
Last Mic	Last Mic	Last Mic	Last Mic

File
Connect
Tools
Help
manual_example
Ethernet Port
Channels
Matrix
Automixer
Wiring
Safety Mute
Expand All
Collapse All
Add Channels...
Remove Channels

There are two styles of automixer groups – gating and gain-sharing. The controls for these two styles of groups are described next.

Gating Automixer Parameters

NOM Limit

NOM Limit specifies the maximum number of microphones that can be gated on for a particular gated automixer group. This does not affect a gain sharing mixer.

Hold Time

Hold Time specifies the amount of time a channel remains active after the last detected significant signal level. This should be set long enough to remain active during short pauses in speech.

Camera Activity Time

Camera Activity Time specifies how long the microphone must be considered active before a camera indicator is set. The camera indicator is a status message that can be used with an external control system to indicate that a particular microphone is active. Shorter times mean the indicator is easier to set based on local talker activity. Longer times mean that it takes longer before the camera gating activity indicator is triggered.

Priority Attenuation

Each automixer group can have a priority attenuation setting, in dB. A value of 0 means the higher priority microphone comes first in the ordering of which microphones to gate on, but does not otherwise attenuate a lower priority microphone. A priority attenuation value greater than 0 causes the lower priority microphones to be attenuated (in addition to any NOM limit effects) by the priority attenuation when a higher priority microphone is active.

Gain-sharing automixer groups can use the priority attenuation to simulate a “soft chairman" priority ducking.

Off Attenuation

Off Attenuation is the amount of attenuation applied to gated channels when they are not active. This should be set high enough that inactive channels don't contribute too much noise and reverberation to the mix. The ideal value for this parameter may increase with the number of microphones in the system. The default value is 15 dB.

Decay Time

Decay time is the amount of time a gated channel takes to ramp its gain down from open (0 dB) to its off attenuation. This should be set long enough to provide a smooth transition as the talker stops speaking.

Gating Indicators

Channel activity status (the gate light for each microphone) is available for microphones regardless of whether they are in a gain sharing automixer group or a gating automixer group. The gating status lights can be useful for output to channel activity LEDs via the logic outputs and control system displays.

Camera activity is similar to channel activity, but has some additional time that the microphone must be gated before the camera gating indicator is made active. The camera gating status is intended to be used with logic outputs or control systems that interface to a camera positioning system that can have various presets according to which microphones are active.

Adaptive Threshold

Adaptive threshold is the level in dB relative to its noise floor a signal must have to be eligible to be considered active. Higher settings makes the channel less sensitive - harder to turn the microphone on, while lower settings make it more sensitive - easier to turn the microphone on.

Priority

The microphone priority parameter can be used with gated automixer groups to provide a priority of which microphones to keep gated on when the NOM limit is reached and can also provide a ‘soft chairman’ functionality by prioritizing which microphones can be gated on. Microphones with priority 1 are the highest priority, microphones with priority 4 are the lowest priority.

If there is a group NOM limit, the priority parameter helps determine which microphones are allowed to gate on. If the NOM limit is reached, a new high priority microphone turns off a lower priority microphone to make room for itself (if a lower priority microphone is currently on). If all of the open microphones have the same priority, they operate on a first come, first served basis. In addition to the NOM limit sequencing, some attenuation may be applied to lower priority microphones when a higher priority microphone becomes active.

Chairman Mic

The chairman mic feature allows the activation of microphones of important talkers to suppress activation of other microphones. Each microphone may be individually configured as chairman or non-chairman. Multiple microphones in the same group may be configured as chairman mics. If a chairman mic is activated, all non-chairman mics in its automixer group is off-attenuated. Other chairman mics, however, would still be allowed to activate.

Last Mic Mode

When using the gated automixer, last mic on mode can be selected individually for each virtual channel. Depending on which channels have last mic on enabled, the behavior may differ. Last mic on mode is ignored when using the gain sharing mixer.

- If no microphones have last mic mode enabled, all of the channels gate off when no channels are active
- If all of the microphones have last mic mode enabled, the last mic to have any activity is always gated on.
- If only one microphone has last mic mode enabled, this microphone turns on when no other microphones are active. An example of this could be with an instructor’s microphone.
- If some microphones have last mic mode on and some do not, then the behavior varies depending on whether the last active microphone has last mic mode on. If so, that microphone is enabled, if not, then the first microphone in the group with last mic mode on is enabled.

Gain Sharing Automixer Parameters

Slope

The Slope parameter determines the selectivity of how the gain is adjusted on the gain-sharing automixer by setting a multiplier on the gain that is applied to active microphones. The difference in levels detected by the automatic microphone on the active microphones are scaled by the slope parameter to create a gain for the automixer. For systems with large numbers of microphones, increasing the slope biases the system to provide gain to the more active microphones. The default value is 2.

Channel Bias

The channel bias control allows the automixer to be biased towards (positive bias value) or against (negative bias value) activating a particular microphone more so than other microphones. When the channel bias is positive, the signal that the automixer sees is made louder by the gating bias value than it really is, even though the actual signal level is unchanged.

An application for channel bias is when there are wireless presenter microphones that are also reinforced into the local room in addition to other microphones that are not reinforced into the room. The wireless microphones can be biased to become active even if the presenter gets close to another microphone – this keeps the reinforcement heard in the local room and not change the tonality as could happen if another microphone became active.

Processing Delay

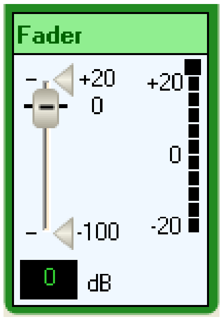
The delay processing allows the designer to add up to 1000 milliseconds of delay on the input channels. While the delay is set in milliseconds in the user interface, it can be manually set through the command console in samples where each sample represents 1/48 of a millisecond.

The input delay may be enabled and disabled and may be adjusted from 0 to 1000 msec.

Controlling Fader

The fader control enables the user to add gain or attenuate the input signal from +20 dB to -100 dB in 0.1 dB increments. This gain or attenuation is applied in the digital domain. The fader control is shown in the following figure.

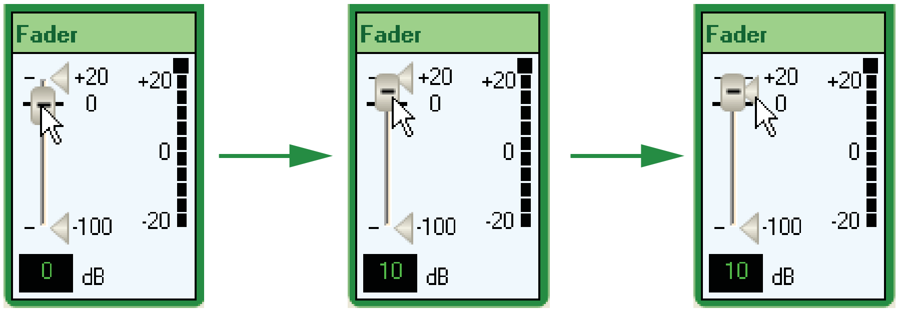
Fader Control in SoundStructure Studio



A maximum and minimum gain range can be specified for the input faders to make it possible to limit user gain control by moving the triangles associated the gain slider. To set the maximum fader gain, adjust the main slider to the desired maximum gain and then move the upper triangle to that level.

Similarly, to adjust the minimum gain, adjust the main slider to the desired minimum level and them move the lower triangle to that location. The steps to set the maximum fader gain to +10 dB are shown in the following figure.

Adjusting the Maximum Fader Gain



See Creating Advanced Applications for an application where the user minimum and maximum faders have been used.

It is recommended that any user adjustment of gain control for input signals control the input fader. This allows the analog input gain (not the fader) to be used for calibration of the input device to the SoundStructure device to ensure the input reaches the 0 dBu nominal signal level of the SoundStructure device. The fader can then be used to make additional adjustments. This ensures when the fader is set back to 0 dB that the analog input gain is still properly calibrated for the connected device.

The signal level meter next to the gain fader shows the signal activity after all the input processing is applied. If an input is muted, the signal level meter for the fader shows no signal activity.

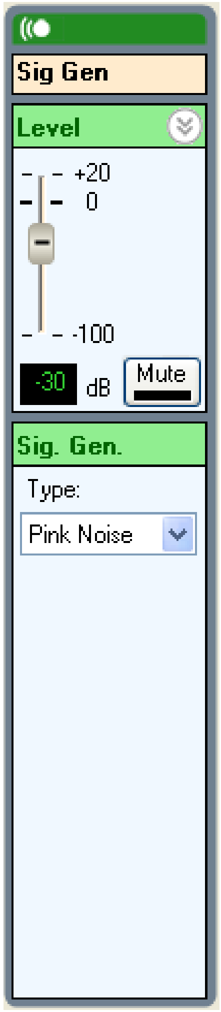
See Installing SoundStructure Devices for additional information on setting signal levels.

Defining a Signal Generator

Each SoundStructure device can have a single signal generator defined which can generate white noise, pink noise, a sine wave, and a sine sweep. By default, each project has a signal generator with pink noise at a level of -30dB added to the project.

The user control of the signal generator is shown in the following figure. The type of noise is selected from the Type pull-down control.

Signal Generator User Controls in SoundStructure Studio

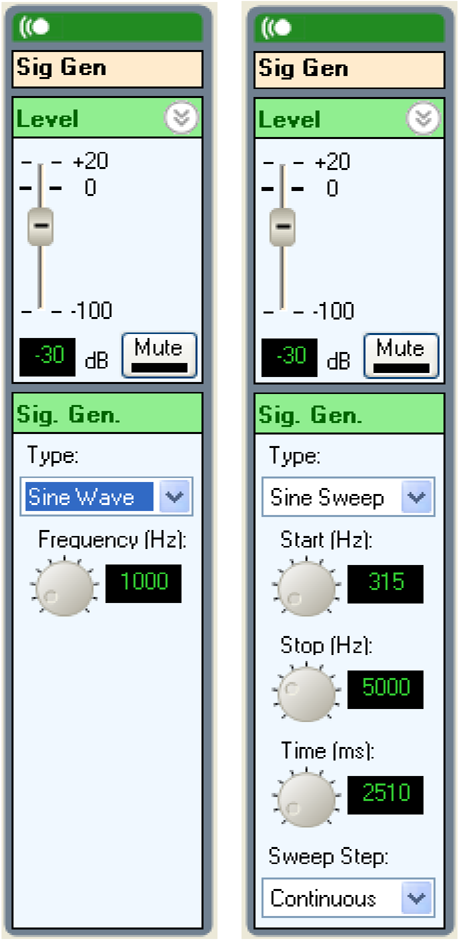


Type: Pink Noise

To use the signal generator, unmute the crosspoint at the signal generator to the desired outputs. Typically the signal generator is routed to loudspeakers as part of the setup process (see Installing SoundStructure Devices) to ensure loudspeakers are active and to adjust the loudspeaker levels in the room.

The controls for sine and sine sweep allow for additional parameters to be set, as shown in the following figure.

Sine and Sweep User Controls in SoundStructure Studio



-30 dB

Type: Sine Wave

Frequency (Hz): 1000

-30 dB

Type: Sine Sweep

Start (Hz): 315

Stop (Hz): 5000

Time (ms): 2510

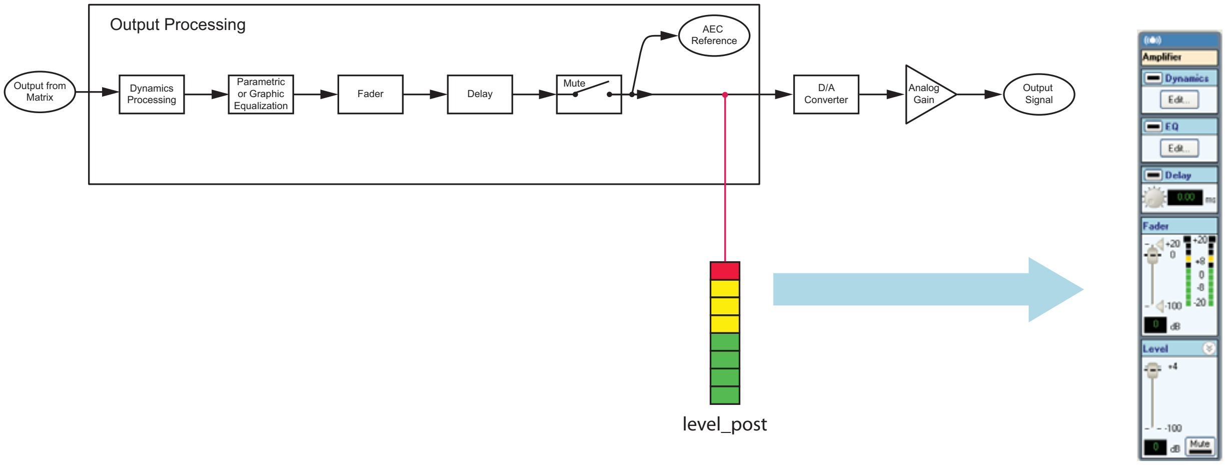
Sweep Step: Continuous

Setting Output Signals

This section describes the user interface for setting output signals. Every output signal has the processing capabilities described in the following section.

All output signals have signal meters, as shown in the following figure. To enable the signal meters, select **Tools** > **Options**. Choose the meters entry and select **Enable Meters**. Meters may also be enabled by right clicking on the meter indicator on the lower right portion of the main SoundStructure Studio window. This figure shows the `level_post` meter for an output and the SoundStructure Studio user control for the meters and gain control. As presented in Installing SoundStructure Devices, the level slider affects the analog signal level on the output of the digital to analog converter. Positive gain is added in the digital domain and is shown in the signal meter, negative gain is implemented in the analog domain and not shown on the signal meter.

Output level_post Meter Signal Meter User Controls in SoundStructure Studio


Output Processing
AEC Reference
Output from Matrix
Dynamics Processing
Parametric or Graphic Equalization
Fader
Delay
Mute
D/A Converter
Analog Gain
Output Signal
level_post
Amplifier
Dynamics
Edit...
EQ
Edit...
Delay
0.00
ms
Fader
+20
0
-100
+20
+8
0
-8
-20
0
dB
Level
+4
-100
0
dB
Mute

Processing Output Dynamics

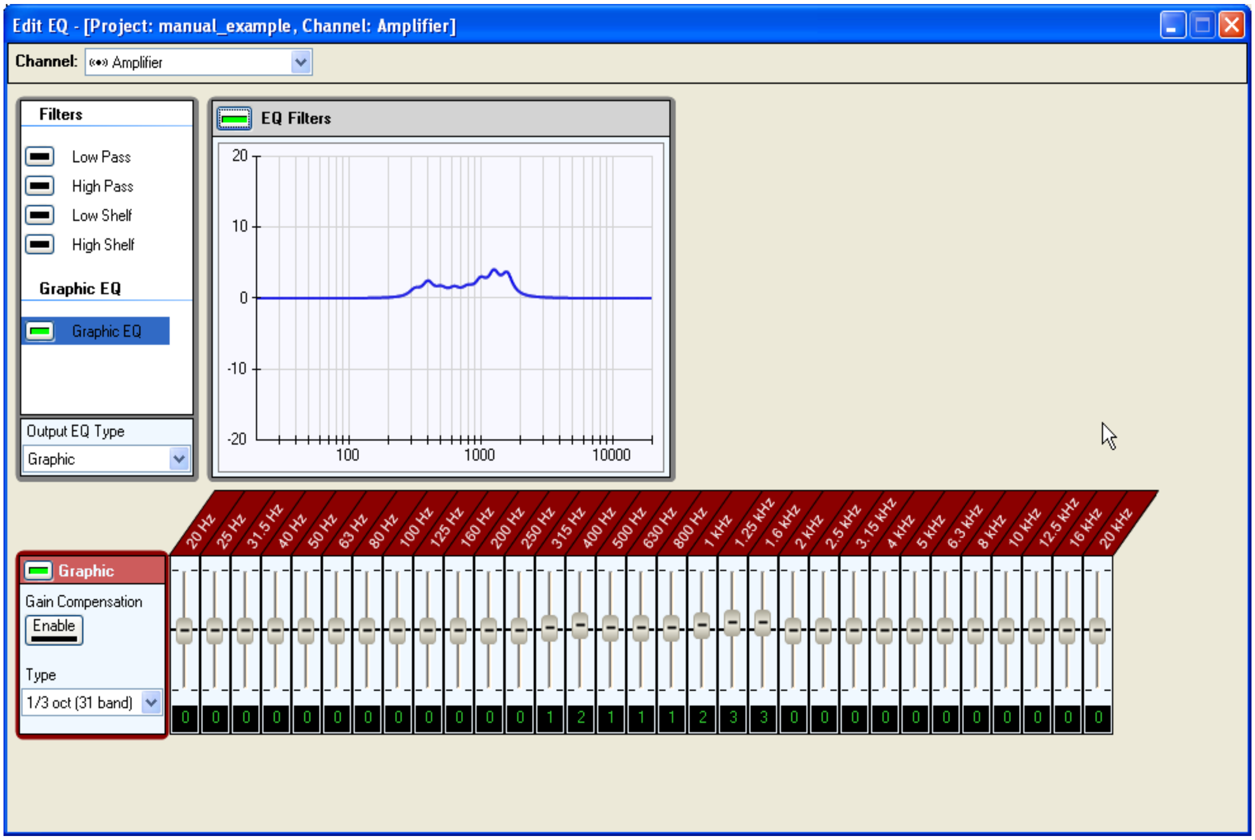
The output dynamics processing available on the outputs is the same as the input dynamics processing described previously in the Using Dynamics Processors section of Setting Input Signals in this chapter.

Processing Output Equalization

The output equalization includes a dedicated Low Pass, High Pass, Low Shelf, and High Shelf filter. In addition the designer may enable either 10 bands of parametric equalization (the same as the input processing) or an octave, 2/3 octive, or 1/3 octave graphic equalizer.

To enable the graphic equalizer, select **Graphic** from the Output EQ Type parameter and to enable the parametric equalizer, select **Parametric** from the Output EQ Type parameter, as shown in the following figure.

Output EQ Type Parameter in SoundStructure Studio


Edit EQ - [Project: manual_example, Channel: Amplifier]

Channel: Amplifier

Output EQ Type: Graphic

Gain Compensation: Enable

Type: 1/3 oct (31 band)

Filters
Low Pass
High Pass
Low Shelf
High Shelf
Graphic EQ
Graphic EQ
EQ Filters
20
10
0
-10
-20
100
1000
10000
Graphic
20 Hz
25 Hz
31.5 Hz
40 Hz
50 Hz
63 Hz
80 Hz
100 Hz
125 Hz
160 Hz
200 Hz
250 Hz
315 Hz
400 Hz
500 Hz
630 Hz
800 Hz
1 kHz
1.25 kHz
1.6 kHz
2 kHz
2.5 kHz
3.15 kHz
4 kHz
5 kHz
6.3 kHz
8 kHz
10 kHz
12.5 kHz
16 kHz
20 kHz
0 0 0 0 0 0 0 0 0 0 0 0 1 2 1 1 1 2 3 3 0 0 0 0 0 0 0 0 0 0 0

The center frequencies of a graphic equalizer are specified in the ISO 266 standard. These are similar to the standard set of resistor values, but the series is chosen to map well to fractional octave and decade intervals between center frequencies. The nominal frequencies are used to label each band in the equalizer.

Depending on the fractional octave size of the equalizer, a different number of bands is needed to cover the audio frequency range. The most common graphic equalizers (and those implemented in this algorithm) are 1-octave (10 band), 2/3-octave (15 band), and 1/3-octave (31 band).

The nominal and exact center frequencies of these equalizers are shown in the following table.

Equalizer Nominal and Center Frequencies

Center Freq (Hz)	1 octave band	2/3 octave band	1/3 octave band
20	—	—	0
25	—	0	1
32	0	—	2
40	—	1	3
50	—	—	4
63	1	2	5

Equalizer Nominal and Center Frequencies

Center Freq (Hz)	1 octave band	2/3 octave band	1/3 octave band
80	—	—	6
100	—	3	7
125	2	—	8
160	—	4	9
200	—	—	10
250	3	5	11
315	—	—	12
400	—	6	13
500	4	—	14
630	—	7	15
800	—	—	16
1.000	5	8	17
1.250	—	—	18
1.600	—	9	19
2.000	6	—	20
2.500	—	10	21
3.150	—	—	22
4.000	7	11	23
5.000	—	—	24
6.300	—	12	25
8.000	8	—	26
10.000	—	13	27
12.500	—	—	28
16.000	9	14	29
20.000	—	—	30

These band edges are exactly between the center frequencies. At the band edges, the gain of the equalizer band is half the gain (in dB) at the center frequency. Adjacent bands in the graphic equalizer bleed over into each other and affect each others' total gain which can increase the amount of time a user must spend adjusting the equalizer to arrive at a desired frequency response. The graphic equalizer provides a gain compensation control that corrects the gain settings of each band to provide the desired gain specified by the user at each center frequency.

Processing Delay

The delay processing allows the designer to add from 0 to 1000 milliseconds of delay on the output channels. While the delay is set in milliseconds in the user interface, it can be manually set through the command API in samples where each sample represents 1/48 of a millisecond.

Processing Submix Signals

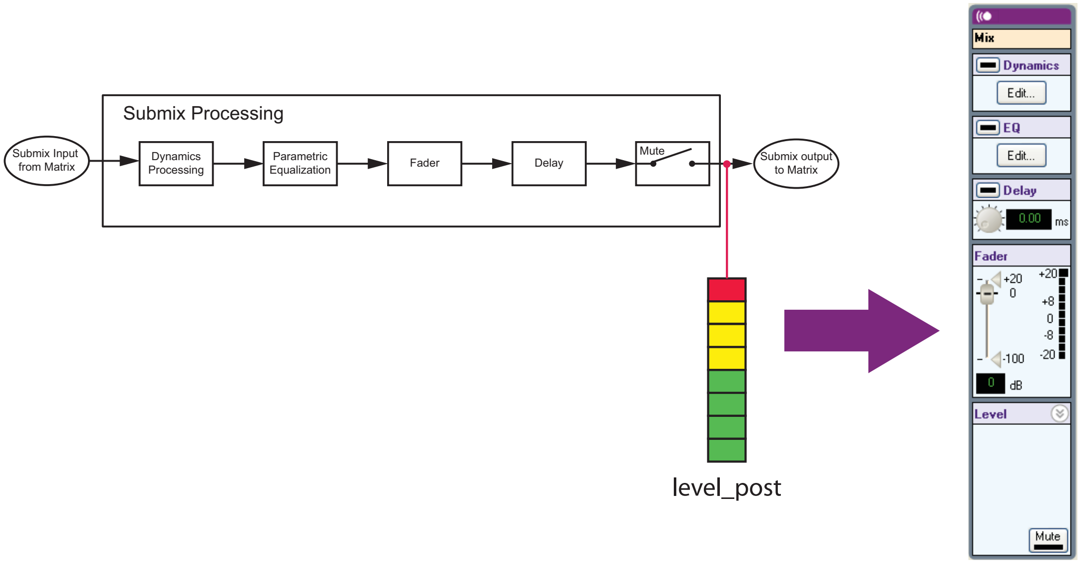
This section describes the processing that is available for each submix channel. Submixes may be defined as mono virtual channels or stereo virtual channels. When the submix is a stereo virtual channel, the processing is applied equally to both the left and the right physical channels that define the stereo virtual channel. Each time a signal is sent to a submix and received back into the matrix, 1.5 msec is added to the delay of the signal.

Note: Submix Signal Delay

Routing a signal to a submix adds 1.5 milliseconds of delay to the signal.

The submix processing flow is shown in the following figure along with the location of the submix signal level meter. The gain on the submix can be adjusted with the fader control.

Submix Processing Flow and Submix Signal Level Meter


Submix Processing
Submix Input from Matrix
Dynamics Processing
Parametric Equalization
Fader
Delay
Mute
Submix output to Matrix
level_post
Mix
Dynamics
Edit...
EQ
Edit...
Delay
0.00
ms
Fader
+20
0
-100
+20
+8
0
-8
-20
0
dB
Level
Mute

Processing Output Dynamics

The output dynamics processing available on the outputs is the same as the input dynamics processing and is described in the Using Dynamics Processors section.

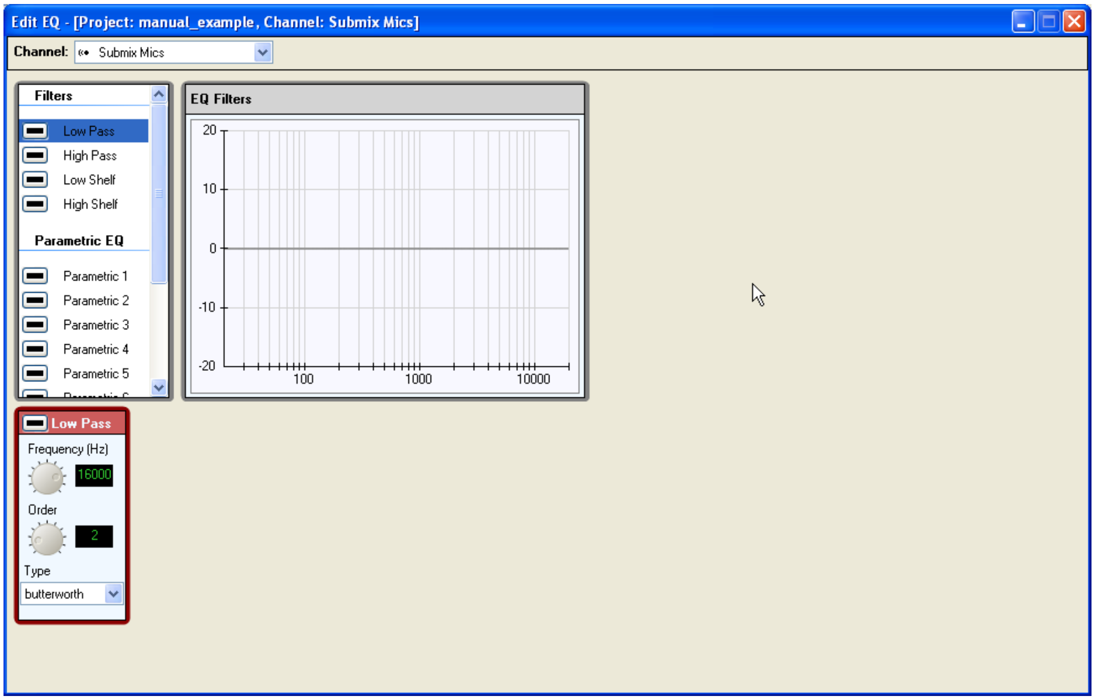
Processing Output Equalization

As shown in the following figure, the equalization processing that is available for each submix consists of a dedicated list of the following:

- Low Pass
- High Pass
- Low Shelf
- High Shelf

- 10 parametric equalizers

Submix Equalization Process


Filters
Low Pass
High Pass
Low Shelf
High Shelf
Parametric EQ
Parametric 1
Parametric 2
Parametric 3
Parametric 4
Parametric 5
EQ Filters
20
10
0
-10
-20
100
1000
10000
Low Pass

Edit EQ - [Project: manual_example, Channel: Submix Mics]

Channel: Submix Mics

Frequency (Hz) 16000

Order 2

Type butterworth

To enable a filter, click the check box next to the filter. This makes the filter the active filter and allows the parameters to be changed as shown next.

The cut off frequency can be adjusted between 0 Hz and 20,000 Hz, the order can be adjusted from 2nd to 8th, and either a Butterworth or Linkwitz-Riley filter may be selected.

For each of the 1 parametric filters, the designer can choose from:

- Parametric filter
- Notch filter
- Allpass filter

Parametric filters emphasize or de-emphasize the center frequency with a gain and bandwidth setting. The user can specify the bandwidth (in octaves), center frequency (in Hz), and gain (from 0 to 20 dB).

Notch filters eliminate energy (attenuate only) at the center frequency. The amount of attenuation for the signal is determined by the bandwidth (in octaves) selected. The bandwidth is defined as where the gain is -3 dB.

Allpass filters do not modify the gain of the signal, but change the phase. For a second order allpass filter, the phase shift is 0 degrees at 0 Hz, 360 degrees at high frequencies, and 180 degrees at the center frequency. The bandwidth is defined as the bandwidth (in octaves) where the phase shift is 90 degrees and 270 degrees.

Processing Delay

The delay processing allows the designer to add up to 1000 milliseconds of delay on the submix signal. While the delay is set in milliseconds in the user interface, it can be manually set through the command API in samples where each sample represents 1/48 of a millisecond.

Controlling Fader

The fader control enables the user to add gain or attenuate the submix signal from +20 dB to -100 dB with a resolution of 0.1 dB. This gain is applied in the digital domain.

A maximum and minimum gain range can be specified for the submix faders to limit the user gain control. The process of setting the min and max volume controls is described in the Controlling Fader in the Setting Input Signals section.

The signal level meter next to the submix fader shows the signal activity after all the submix processing is applied. If a submix is muted, the signal level meter for the fader shows no signal activity.

Using the Matrix Page

The matrix page is where input virtual channels are routed to output channels through the matrix crosspoints and crosspoint gains.

A typical matrix page is shown in the following figure with the input signals on the left and the output signals across the top. All the unmuted crosspoints are shown as bold and the value of each crosspoint is shown in dB. A bold 0 means that the input signal is routed to the output signal and its amplitude is unchanged.

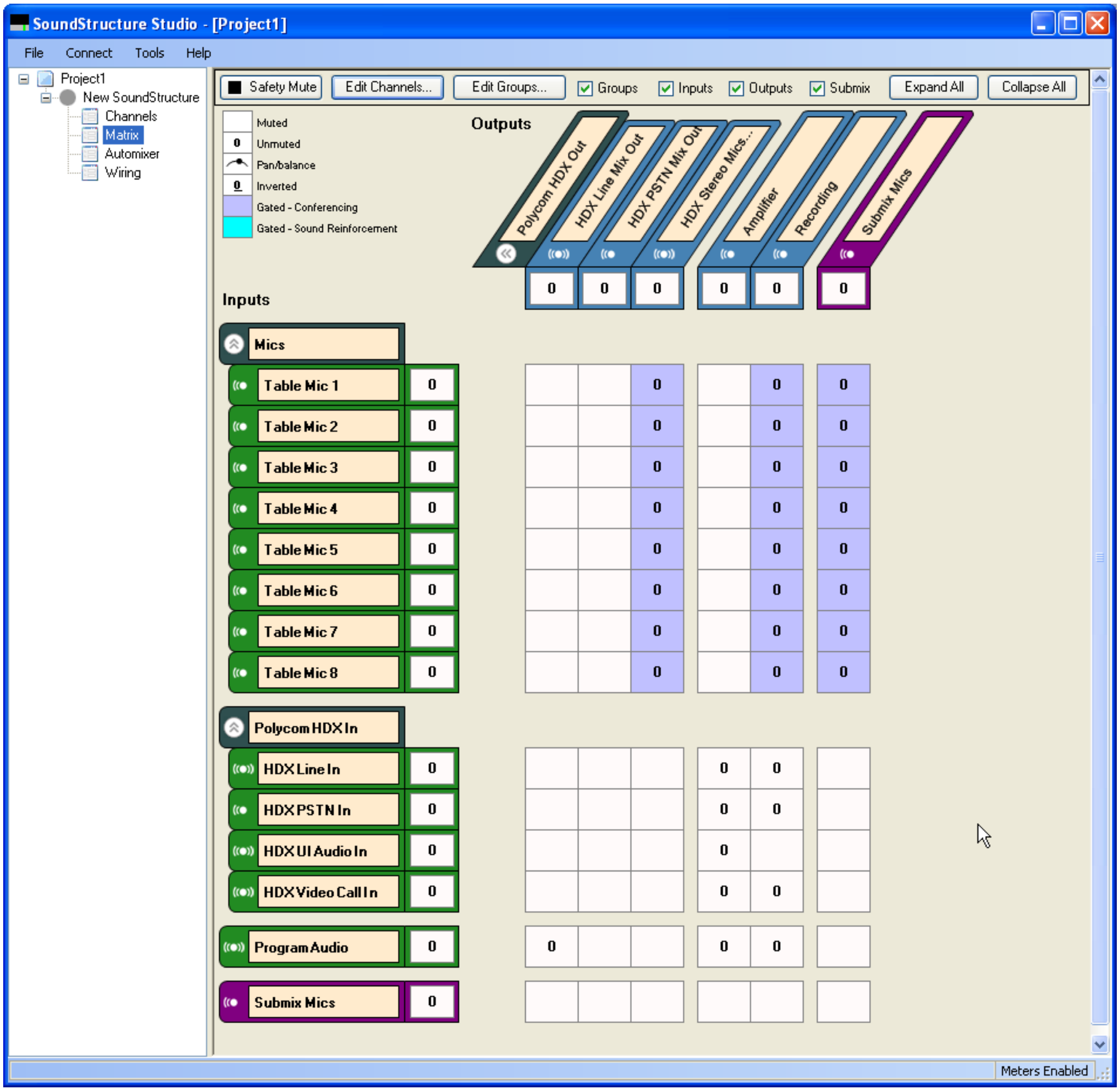
Outputs are created from inputs by summing the values in the column associated with each output signal.

Since input and output channels may be either monaural or stereo virtual channels, there are two special cases to consider when setting crosspoint values:

1. When a stereo input channel is mapped to a mono output channel with a gain of 0 dB, the left and right physical channels are automatically attenuated by 3 dB to create the mono output. The 3 dB attenuation value is used because it is assumed the left and right signals are uncorrelated.
2. When a mono input signal is mapped to a stereo output signal with a gain of 0 dB, the mono input is mapped to both the left and the right physical output channels with an attenuation of 3 dB.

Each group of virtual channels has a heading associated with it - the virtual channel group name - that allows the group to be collapsed or expanded.

.Matrix Page in SoundStructure Studio

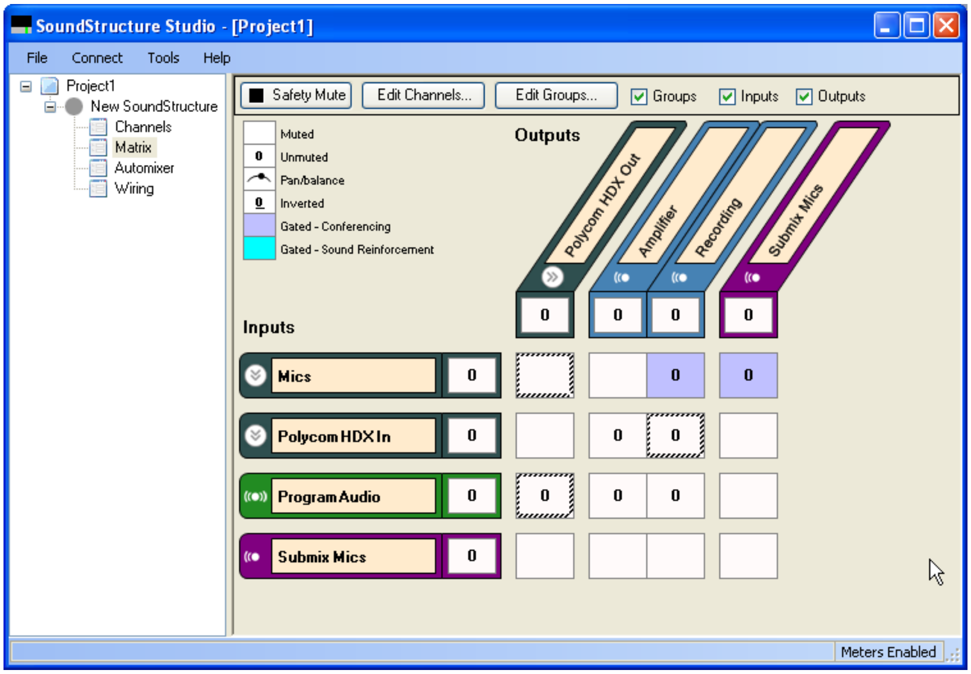

SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
New SoundStructure
Channels
Matrix
Automixer
Wiring
■ Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Expand All
Collapse All
Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Polycom HDX Out
Inputs
Mics
Polycom HDX In
Meters Enabled

Inputs		HDX Line Mix Out	HDX PSTN Mix Out	HDX Stereo Mics...	Amplifier	Recording	Submix Mics
		0	0	0	0	0	0
Table Mic 1	0			0		0	0
Table Mic 2	0			0		0	0
Table Mic 3	0			0		0	0
Table Mic 4	0			0		0	0
Table Mic 5	0			0		0	0
Table Mic 6	0			0		0	0
Table Mic 7	0			0		0	0
Table Mic 8	0			0		0	0
HDX Line In	0				0	0	
HDX PSTN In	0				0	0	
HDX UI Audio In	0				0		
HDX Video Call In	0				0	0	
Program Audio	0	0			0	0	
Submix Mics	0						

The virtual channel groups may also be collapsed to create a matrix that looks like the one in the following figure. The collapsed group crosspoints shows the underlying values of the individual crosspoints if all the values are the same. For crosspoints whose value differs for members in the group, a shaded boundary is shown. This can be seen in the matrix crosspoint of the Codec In group to the Phone Out virtual channel.

The collapsed view simplifies the configuration and setup of the system as there are fewer crosspoints to manage.

Collapsed View of Matrix Page in SoundStructure Studio


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
New SoundStructure
Channels
Matrix
Automixer
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
Muted
Unmuted
Pan/balance
Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs
Meters Enabled

Inputs		Polycom HDX Out	Amplifier	Recording	Submix Mics
		0	0	0	0
Mics	0			0	0
Polycom HDX In	0		0	0	
Program Audio	0	0	0	0	
Submix Mics	0				

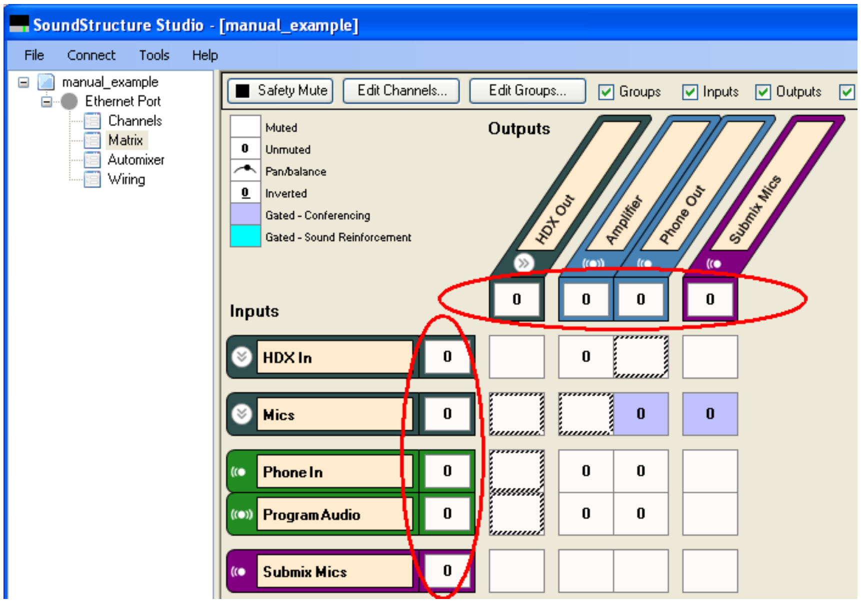
Adjusting Crosspoints

Any matrix crosspoint may be adjusted over the range of +20 dB to -100 dB in 0.1 dB increments.

A maximum and minimum gain range can be specified for the matrix crosspoints to limit the user gain control. The process of setting the min and max matrix gain controls is described in the Controlling Fader section.

The matrix also shows the input or output fader control and mute status for the input and output signals as highlighted in the following figure. The faders and mute status may be adjusted on the matrix page or on the Channels page.

Input and Output Fader Controls and Mute Status on the Matrix Page


SoundStructure Studio - [manual_example]
File
Connect
Tools
Help
manual_example
Ethernet Port
Channels
Matrix
Automixer
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
Muted
Unmuted
Pan/balance
Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
HDX Out
Amplifier
Phone Out
Submix Mics
Inputs
HDX In
Mics
Phone In
Program Audio
Submix Mics

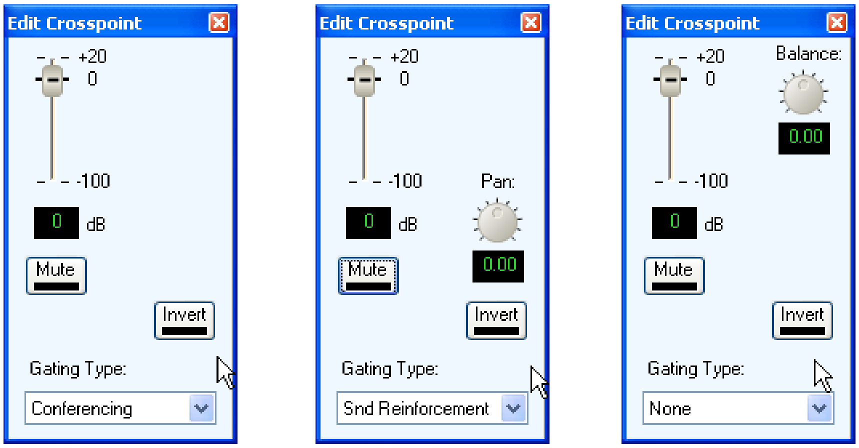
To edit a crosspoint, double left click on the crosspoint to bring up the Edit Crosspoint control. Once the edit crosspoint control is opened, the crosspoint control always goes to its last position. After adjusting a crosspoint, other crosspoints may be changed - without closing the edit crosspoint dialog - by left clicking on the new crosspoint.

Multiple crosspoints can be adjusted simultaneously by pressing the control key at the same time the matrix crosspoints are selected.

As shown in the previous figure, there may be different controls available on the edit crosspoint control depending on the type of input virtual channel and output virtual channel. The following figure shows three crosspoint controls - the first with a mono input to a mono output, the second with a mono input to a stereo output, and the third with a stereo input to a stereo output.

All the Edit Crosspoint controls allow the user to adjust the crosspoint gain in dB by adjusting the slider or by clicking in the value cell and typing in a gain adjustment directly.

Edit Crosspoint User Controls



0 dB

Gating Type: Conferencing

0 dB

Pan: 0.00

Gating Type: Snd Reinforcement

Balance: 0.00

0 dB

Gating Type: None

Muting the Matrix Crosspoint

The matrix crosspoint is muted by clicking the Mute button. Muted crosspoints are shown in the matrix as grayed out values if the Hide Muted Matrix Crosspoints option is not enabled in the Options... selection under the Tools menu. Otherwise if the Hide Muted Matrix Crosspoints is enabled, the muted crosspoints are blank.

Inverting the Matrix Crosspoint

The matrix crosspoint may be inverted meaning that the signal is adjusted by the matrix crosspoint value and negated. The invert feature is there to allow matrix “subtraction” in addition to the more common summing of signals to create output signals.

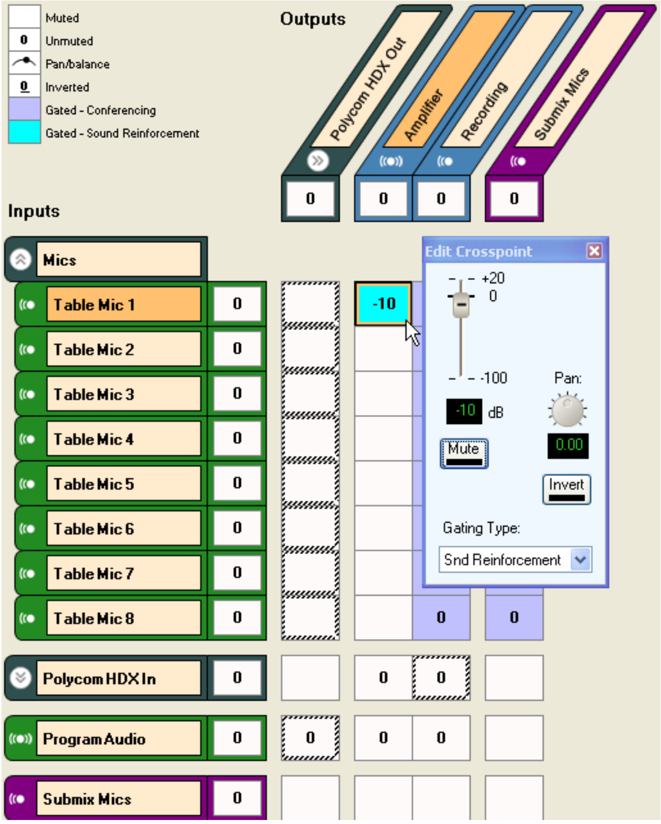
Using the Input Processing Path

When input channels are used in the matrix, there are three possible versions of the input that may be used at the crosspoint: the ungated/recording version, the conferencing version (on C-series or noise canceled on SR-series), and the sound reinforcement version. If the ungated/recording version is selected, the channels page Ungated Type control selects which version of the ungated channel is used.

The selection of which type of input processing to use in the matrix is performed with the matrix crosspoint control as described in the next section.

To select the sound reinforcement version of the input processing, double click the matrix crosspoint to adjust and select Gated and Snd Reinforcement. The crosspoint cell shading changes to light blue to indicate that the sound reinforcement version of the crosspoint is selected. Typically when microphones are sent to loudspeakers, the sound reinforcement version of the input processing should be selected.

Input Processing Path User Controls


Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Polycom HDX Out
Amplifier
Recording
Submix Mics
Inputs
Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
Polycom HDX In
Program Audio
Submix Mics
Edit Crosspoint
+20
0
-100
Pan:
dB
Mute
Invert

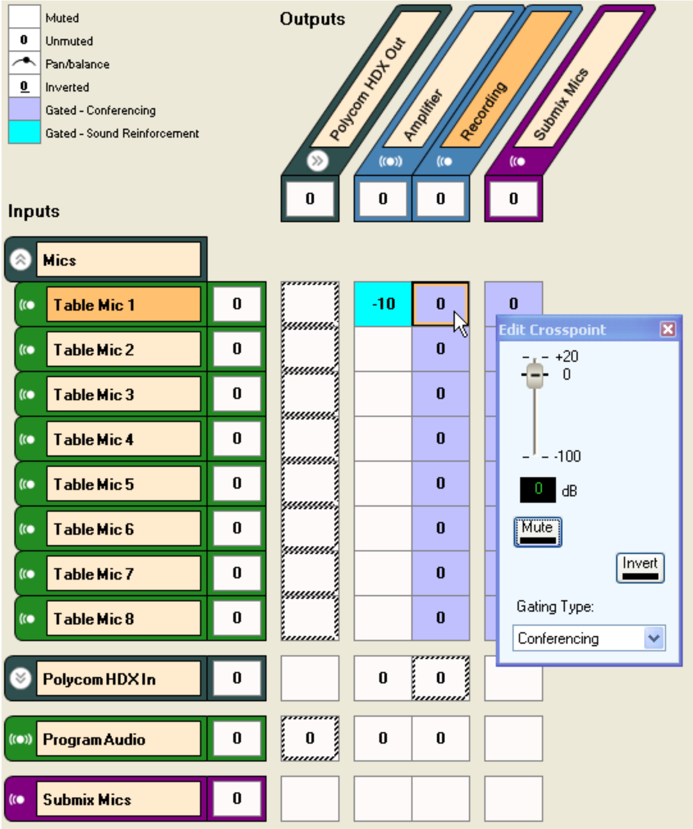
-10 dB

Pan: 0.00

Gating Type: Snd Reinforcement

To select the conferencing version of the input processing, select **Gated** and **Conferencing** as shown in the following figure. The crosspoint background turns blue to indicate the conferencing version of the input processing is selected.

Gated and Conferencing Input Processing Version


Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Polycom HDX Out
Amplifier
Recording
Submix Mics
Inputs
Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
Polycom HDX In
Program Audio
Submix Mics
-10
Edit Crosspoint
+20
0
-100
dB
Mute
Invert

Gating Type: Conferencing

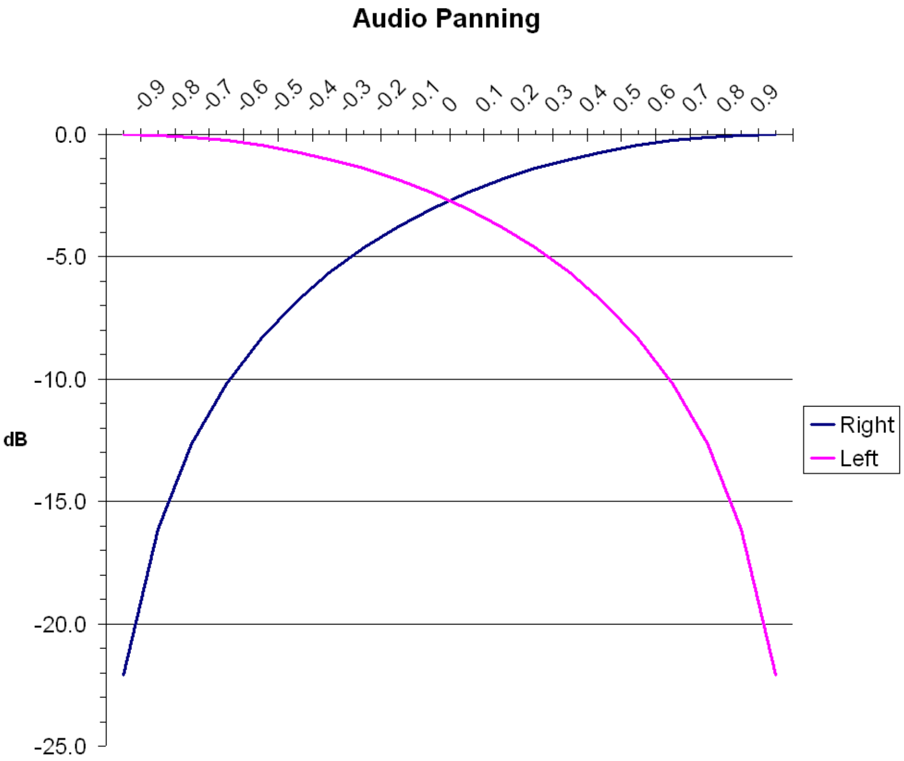
To select the ungated/recording version of the crosspoint, select the None gated version of the input processing. The background of the crosspoint turns white to indicate that the ungated/recording version of the input processing is selected.

Controlling Pan

The pan control enables you to customize how a monaural virtual channel is mapped to a stereo virtual channel. A pan value of 0 means that the monaural input virtual channel is attenuated by 3 dB and sent to both the left and right output channels. The gain (or attenuation) of the matrix crosspoint is also applied to the input signal as it is mapped to the output signal.

A pan value of 1 means that the mono virtual channel is only mapped to the right output physical channel, a value of -1 means that the mono virtual channel is mapped to the left output physical channel. Values between -1 and 1 are shown in the following figure.

Audio Panning Values



Controlling Balance

The balance control allows the designer to adjust how a stereo input signal is mapped to a stereo output signal. A value of 0 means that the left input channel is sent to the left output channel and the right input channel is sent to the right output channel.

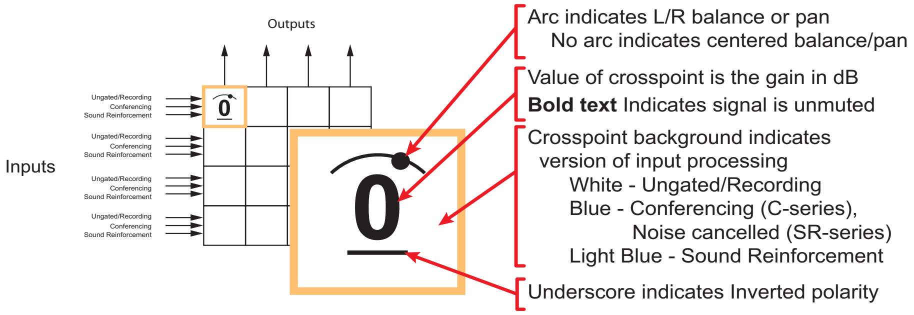
Matrix summary

A summary of the matrix crosspoint visual controls is shown in the following figure and reviewed here.

- Bold values are the gain in dB in the crosspoint.
- An arc with a circle indicates that there is some panning or balance other than the center position in effect.
- An underscore indicates the matrix crosspoint is inverted.

- The background color indicates which version of the input processing is selected - blue indicates the conferencing path (or noise canceled path on the SR-series), light blue indicates the sound reinforcement path, and white indicates the ungated/recording path.

Matrix Crosspoint Visual Controls


Outputs
Inputs
Ungated/Recording
Conferencing
Sound Reinforcement
Ungated/Recording
Conferencing
Sound Reinforcement
Ungated/Recording
Conferencing
Sound Reinforcement
Ungated/Recording
Conferencing
Sound Reinforcement
0
0

Arc indicates L/R balance or pan
No arc indicates centered balance/pan

Value of crosspoint is the gain in dB
Bold text Indicates signal is unmuted

Crosspoint background indicates version of input processing
White - Ungated/Recording
Blue - Conferencing (C-series),
Noise cancelled (SR-series)
Light Blue - Sound Reinforcement

Underscore indicates Inverted polarity

Using the Telephony Channels

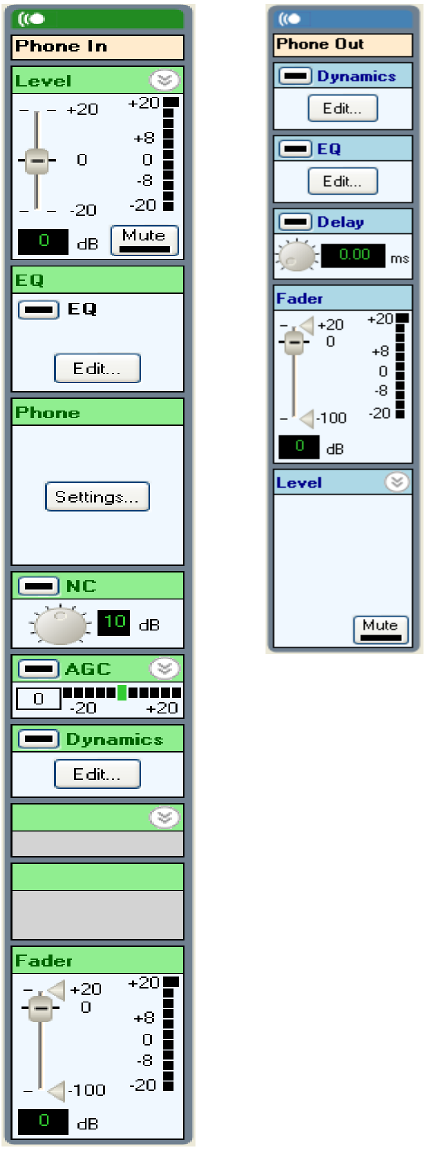
To use a telephone interface, either the SoundStructure TEL1 or TEL2 must be included in the design and installed in the SoundStructure device.

Each telephone interface that is used in the design is represented by two virtual channels: one for the input telephone signal and one for the output telephony signal. An example of these two virtual channels is shown in the following figure.

The signal processing paths for both the input and output channels include equalization, dynamics processing, and audio delay. In addition, the telephone input channel has noise cancellation and automatic gain control that can be applied to the signal received from the telephone line.

The controls for both the telephone input and output channels are described in this section.

Virtual Channel Example

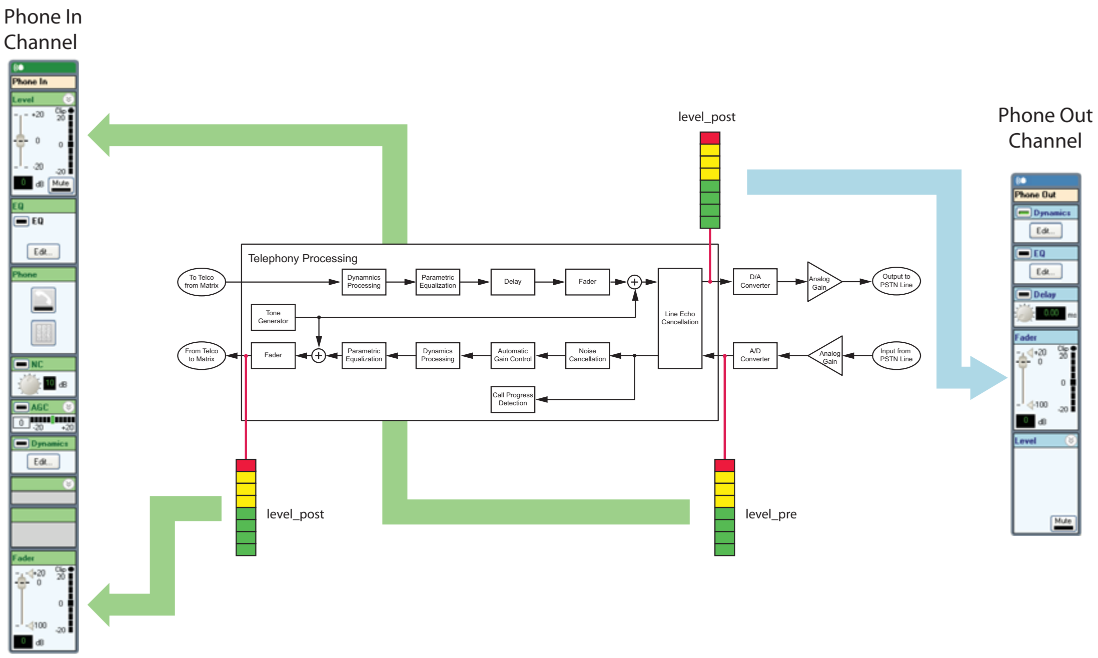

Phone In
Level
+20
0
-20
+20
+8
0
-8
-20
0
dB
Mute
EQ
EQ
Edit...
Phone
Settings...
NC
10
dB
AGC
0
-20
+20
Dynamics
Edit...
Fader
+20
0
-100
+20
+8
0
-8
-20
0
dB
Phone Out
Dynamics
Edit...
EQ
Edit...
Delay
0.00
ms
Fader
+20
0
-100
+20
+8
0
-8
-20
0
dB
Level
Mute

Adjusting Input Gain

The telephone input gain has a range from -20 to +20 dB for adjusting the gain in the analog domain and has a default gain of 0 dB. The gain required depends on the signal levels received from the telephone line. Adjust the telephone gain so that during normal speech there are at least two yellow LEDs lit on the telco receive.

The location of the telco signal meters are shown in the following figure. The input channel meters level_pre meter corresponds to the meter next to the analog input gain adjustment on the telephone input virtual channel. The input channel level_post meter corresponds to the meter next to the input fader control. The output channel `level_post` meter corresponds to the meter next to the output gain adjust.

Telco Signal Meters


Phone In Channel
Phone In
Level
EQ
Edit...
Phone
NC
AGC
Dynamics
Fader
Mute
level_post
Phone Out Channel
Phone Out
Dynamics
EQ
Delay
Fader
Level
Telephony Processing
To Telco from Matrix
Dynamics Processing
Parametric Equalization
Delay
Fader
Line Echo Cancellation
D/A Converter
Analog Gain
Output to PSTN Line
Tone Generator
From Telco to Matrix
Fader
Parametric Equalization
Dynamics Processing
Automatic Gain Control
Noise Cancellation
A/D Converter
Analog Gain
Input from PSTN Line
Call Progress Detection
level_post
level_pre

Processing Noise Cancellation

The noise cancellation processing is available on the telephone input signal. The noise cancellation reduces background noise that is present in the signal that is transmitted from the remote site. The noise cancellation functionality can be turned on or off with the enable button. The amount of noise cancellation can be adjusted from 0 to 20 dB.

The SoundStructure noise cancellation effectively removes different types of background noise ranging from narrow band noise (tones) to broadband noise. For best performance, the noise characteristics should be quasi-stationary, for example, the statistics of the underlying noise are fixed or change slowly over time.

Using Automatic Gain Control (AGC)

Automatic gain control is used to automatically adjust the gain of audio signals so that the average signal level is close to the SoundStructure nominal signal level of 0 dBu. The AGC system is designed to adapt the gain only when valid speech is present.

The AGC can be turned on or off with the AGC enable button. When the AGC is enabled, the current AGC gain (or attenuation) can be viewed from the AGC meter and the text box next to the AGC meter.

The range of the AGC can be adjusted by expanding the AGC control and adjusting the maximum and minimum gains. By default the maximum and minimum gain are set to 6 and -6 respectively on microphone and telephony signals.

The *maximum* AGC value specifies the maximum amount of gain the AGC can apply to increase the input signal level as the AGC tries to reach the SoundStructure nominal signal level.

The *minimum* AGC value specifies the maximum amount of attenuation the AGC can apply to attenuate the input signal as the AGC tries to reach the SoundStructure nominal signal level.

Processing Output Dynamics

The output dynamics processing available on the outputs is the same as the input dynamics processing described in the Using Dynamics Processors section of this chapter.

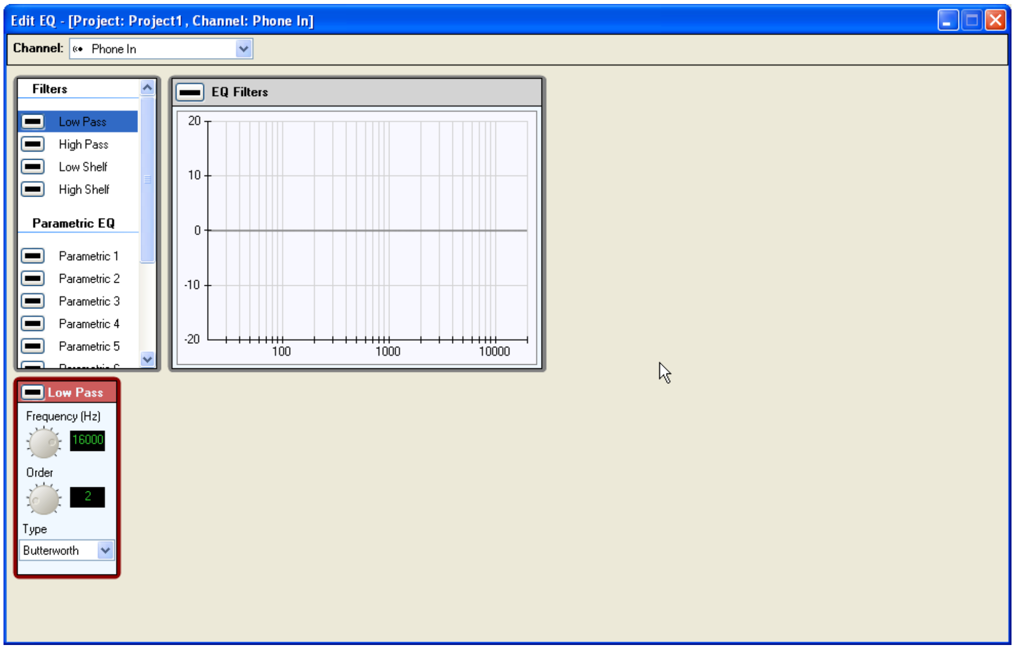
Processing Equalization

The equalization processing that is available for both the telephone input and output signals, as shown in the following figure, consists of the following dedicated filters:

- Low Pass
- High Pass
- Low Shelf
- High Shelf
- 10 parametric equalizers

The telephone input and output can be configured to have different equalization.

Dedicated Equalization Filters


Edit EQ - [Project: Project1 , Channel: Phone In]
Channel:
Phone In
Filters
Low Pass
High Pass
Low Shelf
High Shelf
Parametric EQ
Parametric 1
Parametric 2
Parametric 3
Parametric 4
Parametric 5
EQ Filters
20
10
0
-10
-20
100
1000
10000
Low Pass

Frequency (Hz) 16000

Order 2

Type Butterworth

To enable a filter, click the check box next to the filter. This makes the filter the active filter and allows the parameters to be changed as shown next.

The cut off frequency can be adjusted between 0 Hz and 20,000 Hz, the order can be adjusted from 2nd to 8th, and either a Butterworth or Linkwitz-Riley filter may be selected.

For each of the 1 parametric filters, the designer can choose from:

- Parametric filter
- Notch filter
- Allpass filter

Parametric filters emphasize or de-emphasize the center frequency with a gain and bandwidth setting. The user can specify the bandwidth (in octaves), center frequency (in Hz), and gain (from 0 to 20 dB).

Notch filters eliminate energy (attenuate only) at the center frequency. The amount of attenuation for the signal is determined by the bandwidth (in octaves) selected. The bandwidth is defined as where the gain is -3 dB.

Allpass filters do not modify the gain of the signal, but change the phase. For a second order allpass filter, the phase shift is 0 degrees at 0 Hz, 360 degrees at high frequencies, and 180 degrees at the center frequency. The bandwidth is defined as the bandwidth (in octaves) where the phase shift is 90 degrees and 270 degrees.

Controlling Fader

The fader control enables the user to add gain or attenuate the telephone signal from +20 dB to -100 dB with a resolution of 0.1 dB. This gain is applied in the digital domain.

A maximum and minimum gain range can be specified for the faders to limit the user gain control. The process of setting the min and max volume controls is described in the Controlling Fader section.

There is a fader control on the phone input channel and a fader control on the phone output channel.

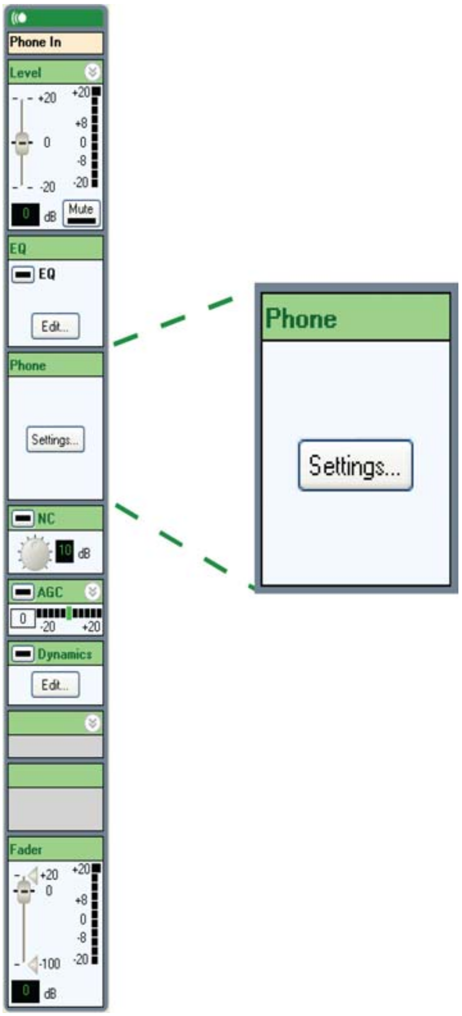
Processing Delay

The delay processing allows the designer to add from 0 to 1000 milliseconds of delay on both the telephone input and output channels. While the delay is set in milliseconds in the user interface, it can be manually set through the command API in samples where each sample represents 1/48 of a millisecond.

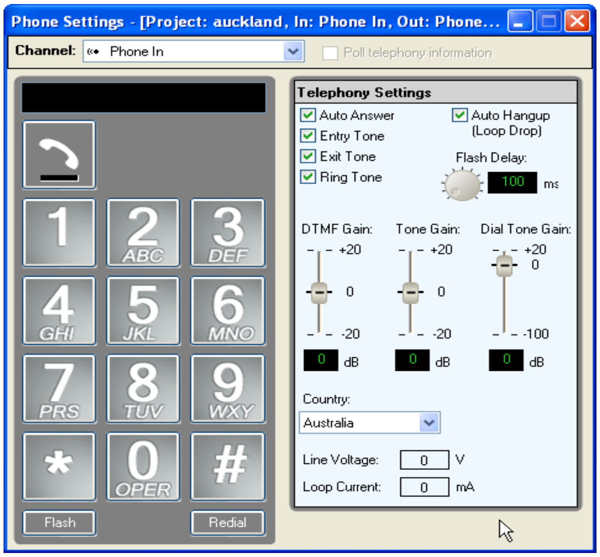
Using Telephone Controls

In addition to the audio processing paths described in this section, telephony channels have additional user controls to configure the telephone interface. Select **Phone Settings** to get access to the telephony specific controls.

Telephony Channel User Controls


Phone In
Level
+20
0
-20
+20
+8
0
-8
-20
0 dB
Mute
EQ
EQ
Edit...
Phone
Settings...
NC
10 dB
AGC
0
-20
+20
Dynamics
Edit...
Fader
+20
0
-100
+20
+8
0
-8
-20
0 dB
Phone
Settings...

The telephony channel controls are shown in the following figure.



Phone Settings - [Project: auckland, In: Phone In, Out: Phone...

Channel: Phone In

☐ Poll telephony information

Telephony Settings

☑ Auto Answer
☑ Entry Tone
☑ Exit Tone
☑ Ring Tone
☑ Auto Hangup (Loop Drop)

Flash Delay: 100 ms

DTMF Gain: 0 dB

Tone Gain: 0 dB

Dial Tone Gain: 0 dB

Country: Australia

Line Voltage: 0 V

Loop Current: 0 mA

Using the Telephone Interface

The telephone interface may be taken off hook by pressing the phone button on the controls page. Once the telephone is off hook, dial digits by pressing the keys on the keypad.

Please note that the telephone must be taken off hook before digits may be dialed. This behavior is different from the Vortex products where dialing digits if the phone were on hook would cause the phone to go off hook. With the SoundStructure products, the phone must be taken off hook prior to dialing.

Enabling Auto Answer

Enabling auto answer sets the SoundStructure device to answer the phone automatically after two rings.

Enabling Entry Tones

Entry tones enabled causes the SoundStructure device to play a short series of tones into the local room to indicate that the phone was answered.

Exit Tone

Enabling exit tones sets the SoundStructure device to play a short series of tones into the local room to indicates that the phone was hung up.

Enabling Ring Tones

Enabling ring tones sets the SoundStructure device to play ring tones into the local room when the telephone line rings.

If Ring Tone is disabled no ring tone is heard although a `phone_ring` status message is generated by the SoundStructure device when the phone is ringing.

Enabling Auto Hang-Up

Enabling auto hang up enabled allows the system to auto hang up based on loop drop detection.

Enabling DTMF Gain

Enabling DTMF gain enables you to adjust the level of the DTMF digits that are played into the local room while dialing the telephone interface.

Adjusting the DTMF gain does not adjust the level of the DTMF digits that are sent to the telephone line.

Using Tone Gain

Tone gain adjusts the level of the tones, including the ring tone, that are played into the local room including the entry and exit tones.

Using Dial Tone Gain

Dial Tone gain adjusts the level of the in room dial tone when the phone is taken off hook.

Using Flash Delay

Flash delay sets the flash timing in milliseconds when the flash feature is executed.

Setting Country Code

The country setting of the telephone interface must be set prior to first use of the telephone line. The country code only needs to be set once to set the appropriate telephone line interface parameters that are region dependent.

Once the country code is set, the phone line may be tested by clicking the phone icon. This takes the selected phone line off-hook. Assuming the signal routing is correct through the matrix, and the phone line is connected and active, dial tone is heard in the local room.

Using Line Voltage and Loop Current

The line voltage and loop current are active whenever the Poll Telephony Information is enabled at the top of the user control. The line voltage and loop current allow for diagnostics of the telephone line. See Appendix A of this manual for more information on how to query the line voltage and loop current values.

Connecting Over Conference Link2

While there are two Conference Link2 (CLink2) interfaces on a SoundStructure device that permit two simultaneous connections to other Polycom devices, only one Polycom Video Codec conferencing system may be connected to a SoundStructure device.

Connecting SoundStructure Conference Link2

As described in the SoundStructure Hardware Installation Guide, each SoundStructure Conference Link2 interface accepts an RJ45 terminated CAT-5e cable. An 18” cable with the proper pin out is provided with each SoundStructure device.

Do not use the Conference Link interface to connect two SoundStructure devices together - the Conference Link interface does not work in that manner. Use the OBAM interface to link multiple SoundStructure devices together.

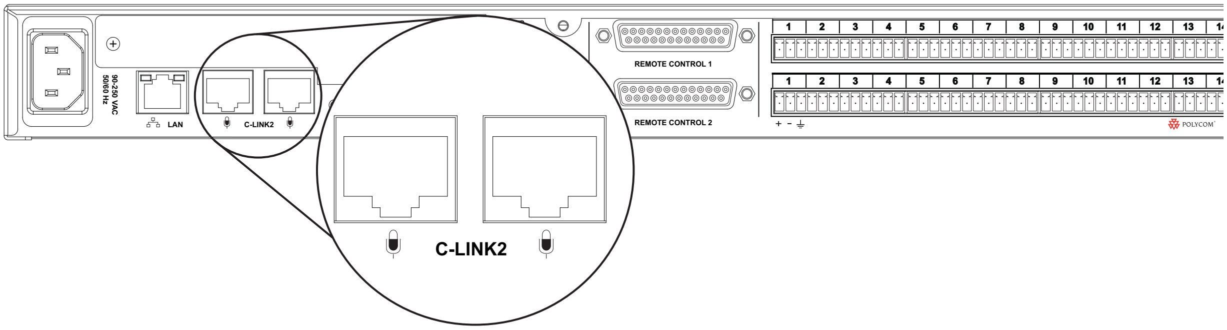
Note: Conference Link2 RJ45 Terminated Cables and Pin Outs

While the Conference Link2 socket accepts RJ45 terminated cables, the pin out is ***not*** the same as the T568A and T568B pin outs that are commonly used with network products. See Specifications or the SoundStructure Hardware Installation Guide for additional cable information including the required pin outs.

The signals that are transmitted between the SoundStructure device and a Polycom Video Codec conferencing system connected over Conference Link2 are kept as digital signals. No analog signals are transmitted between the SoundStructure device or the Polycom Video Codec video conferencing system when connecting to the Polycom Video Codec system with the Conference Link2 interface.

The rear-panel of the SoundStructure product with the Conference Link2 connections highlighted is shown in the following figure.

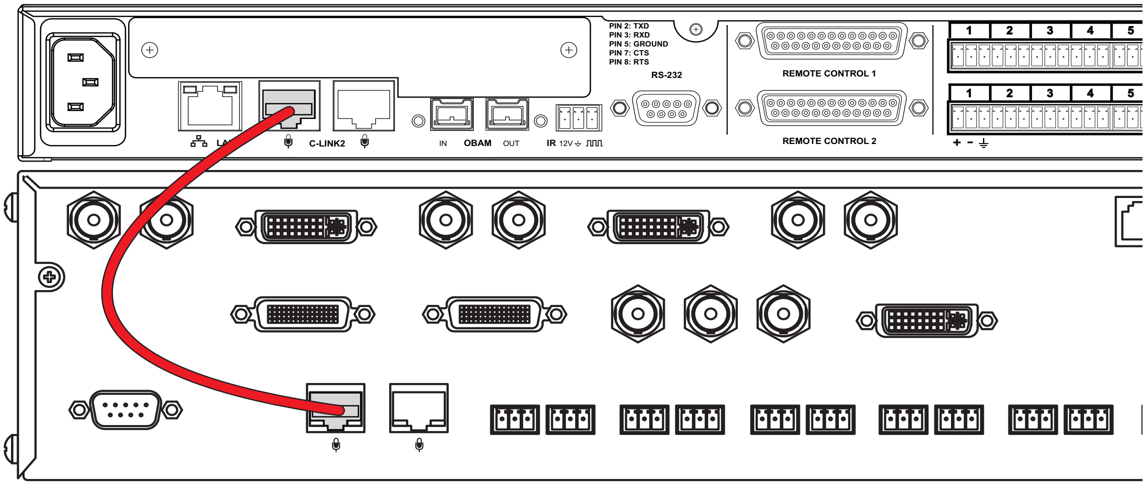
SoundStructure Device Rear-Panel with Conference Link2 Connections


90-250 VAC
50/60 Hz
LAN
C-LINK2
REMOTE CONTROL 1
REMOTE CONTROL 2
1
2
3
4
5
6
7
8
9
10
11
12
13
+ - ⏚
POLYCOM
C-LINK2

Integrating Polycom Video Codec

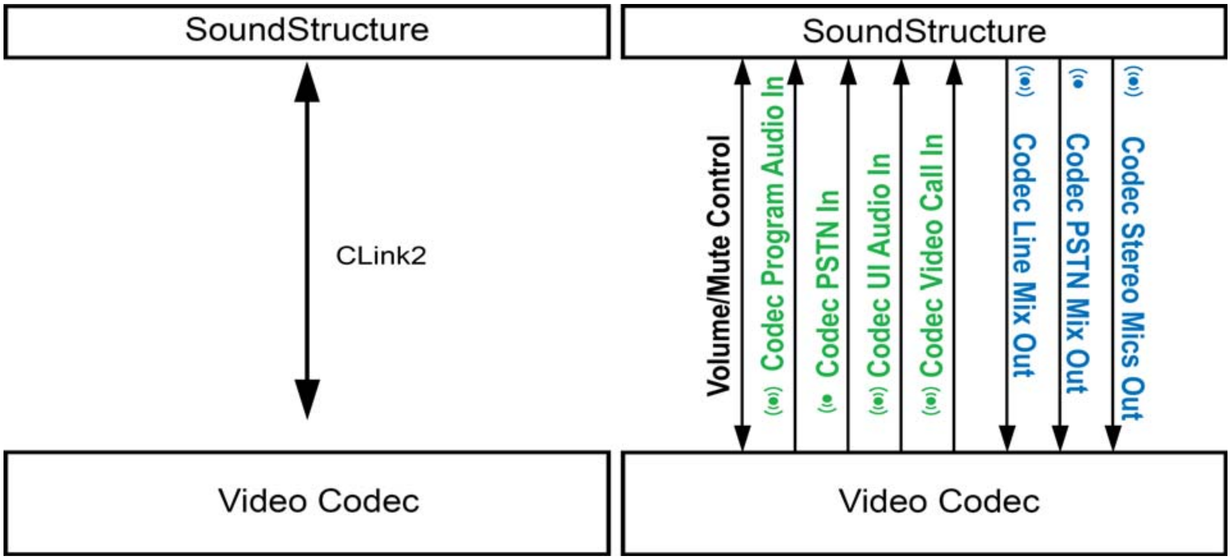
The SoundStructure devices may be connected to the Polycom Video Codec video conferencing system using the supplied Conference Link2 cable as shown in the next figure. Either Conference Link2 port on the SoundStructure device or the Polycom Video Codec system may be used. The Polycom Video Codec system requires firmware release 2.0.1 or higher firmware to be compatible with SoundStructure devices.

Connecting Polycom Video Codecs with SoundStructure Devices using Conference Link2


PIN 2: TXD
PIN 3: RXD
PIN 5: GROUND
PIN 7: CTS
PIN 8: RTS
RS-232
REMOTE CONTROL 1
REMOTE CONTROL 2
C-LINK2
IN
OBAM
OUT
IR 12V
1
2
3
4
5
+ - ⏚

The Conference Link2 interconnect allows for the transmission and reception of multiple digital audio signals between the two devices as shown in the following figures. These signals are described in the following sections.

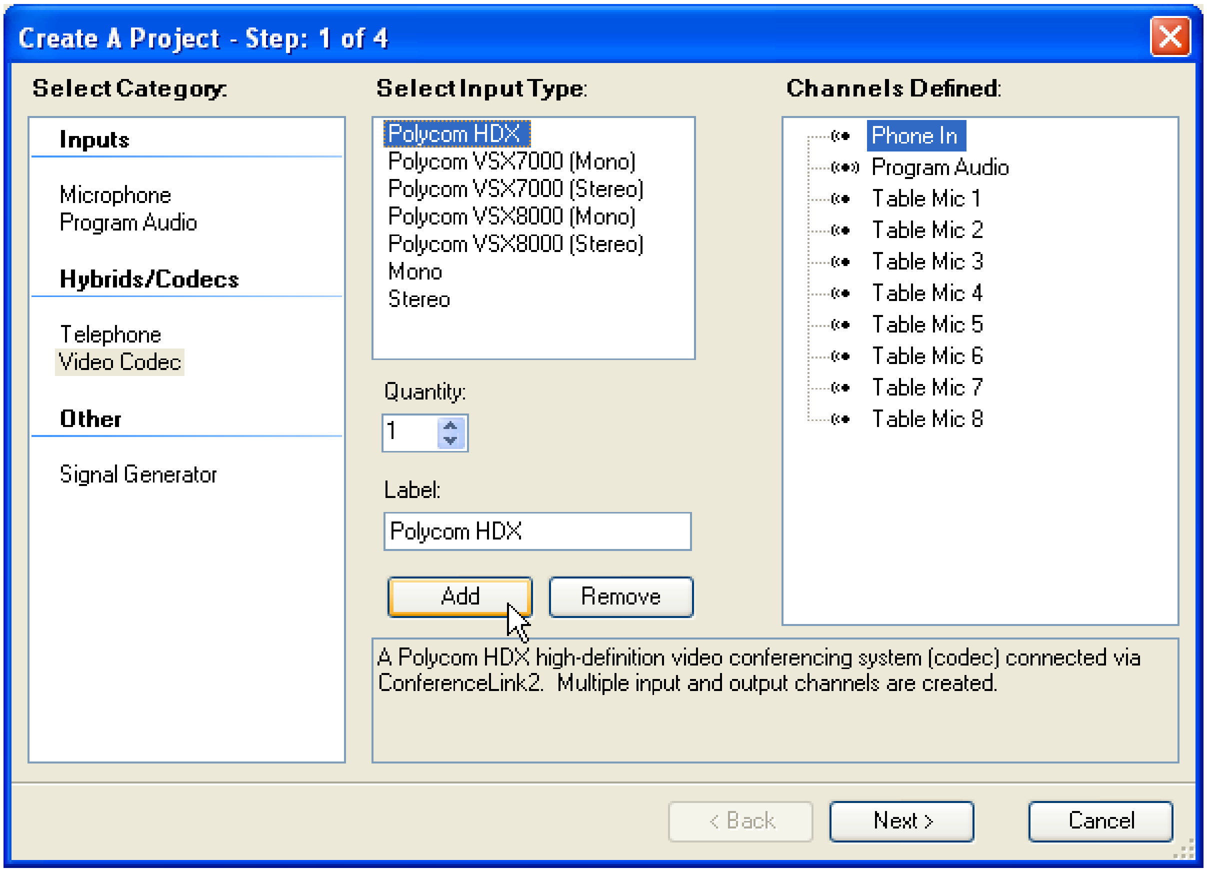
Digital Audio Signals Between SoundStructure Devices and Polycom Video Codecs


SoundStructure
CLink2
Video Codec
SoundStructure
Volume/Mute Control
Codec Program Audio In
Codec PSTN In
Codec UI Audio In
Codec Video Call In
Codec Line Mix Out
Codec PSTN Mix Out
Codec Stereo Mics Out
Video Codec

Designing with The Polycom Video Codec

Within SoundStructure Studio, the Polycom Video Codec system may be selected from the video codec selection category and clicking **Add** to add the codec to the list of inputs as shown in the following figure.

Adding the Polycom Video Codec in SoundStructure Studio


Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Telephone
Video Codec
Other
Signal Generator
Select Input Type:
Polycom HDX
Polycom VSX7000 (Mono)
Polycom VSX7000 (Stereo)
Polycom VSX8000 (Mono)
Polycom VSX8000 (Stereo)
Mono
Stereo
Add
Remove
Channels Defined:
Phone In
Program Audio
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
< Back
Next >
Cancel

Quantity: 1

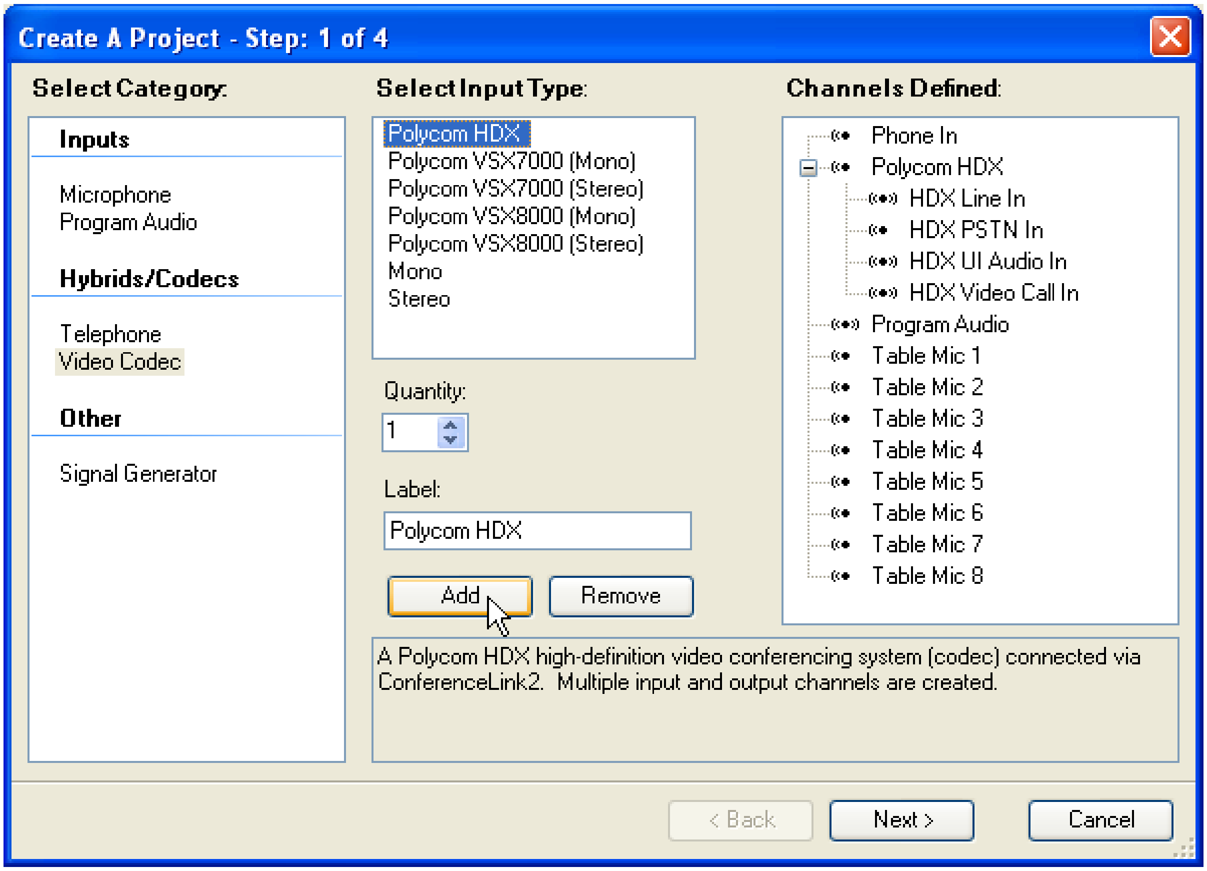
Label: Polycom HDX

A Polycom HDX high-definition video conferencing system (codec) connected via ConferenceLink2. Multiple input and output channels are created.

Editing The Polycom Video Codec Input Channels

After the Polycom Video Codec system is selected, four SoundStructure input virtual channels are automatically added to the input channels as shown in the next figure. If a particular input channel is not going to be used, for example the Codec Voice Call In channel, that channel may be removed from the input channels without affecting the other input channels from the Polycom Video Codec system.

Added Virtual Channels for Polycom Video Codec in SoundStructure Studio


Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Telephone
Video Codec
Other
Signal Generator
Select Input Type:
Polycom HDX
Polycom VSX7000 (Mono)
Polycom VSX7000 (Stereo)
Polycom VSX8000 (Mono)
Polycom VSX8000 (Stereo)
Mono
Stereo
Quantity:
1
Label:
Polycom HDX
Add
Remove
Channels Defined:
Phone In
Polycom HDX
HDX Line In
HDX PSTN In
HDX UI Audio In
HDX Video Call In
Program Audio
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
< Back
Next >
Cancel

A Polycom HDX high-definition video conferencing system (codec) connected via ConferenceLink2. Multiple input and output channels are created.

The input channels from the Polycom Video Codec are described in the following table.

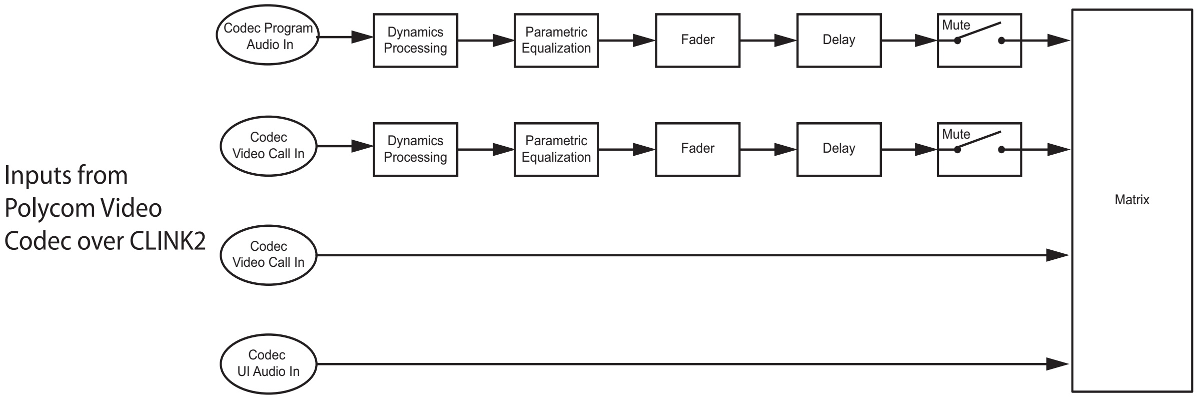
Polycom Video Codec Input Channels

Video Codec Signal to SoundStructure	*Description*
Video Codec Program Audio In	A stereo virtual channel that contains a mix of all non-microphone inputs to the Polycom Video Codec. This audio signal includes the VCR/DVD audio input and the PC audio input. Note that the VCR/DVD and PC audio input are only active when the corresponding video input is selected as a send source for either People or Content video. As an example, the VCR/DVD audio source is only sent to the SoundStructure device when the Video Codec source associated with the VCR/DVD input is selected. If a different video source is selected on the Polycom Video Codec, then this VCR/DVD audio is not sent to the SoundStructure device over the CLink2 interface.
Codec Voice Call In	A mono virtual channel that contains a mono mix of all far-end audio for audio-only calls hosted by the Video Codec. This includes the call on both the PSTN and ISDN voice interfaces.
Video Codec UI Audio In	A stereo virtual channel that contains a mix of all sound effects locally generated by the Video Codec including local ring, ring back, dial tone, boot up audio playback, error tones, and user input audible feedback.
Video Codec Video Call In	A stereo virtual channel that contains a stereo mix of all far-end audio for video calls hosted by the Video Codec. This includes video calls on the ISDN H.320, IP.H323, and IP SIP. If the call is mono, both the left and right channels contain the same audio signal.

Processing The Polycom Video Codec SoundStructure Signals

Each of the signals that the Polycom Video Codec system sends to the SoundStructure device have processing that can be applied as shown in the following figure. This processing is configured through the SoundStructure Studio software.

Processing Signals for Polycom Video Codec


Inputs from Polycom Video Codec over CLINK2
Codec Program Audio In
Dynamics Processing
Parametric Equalization
Fader
Delay
Mute
Codec Video Call In
Dynamics Processing
Parametric Equalization
Fader
Delay
Mute
Codec Video Call In
Codec UI Audio In
Matrix

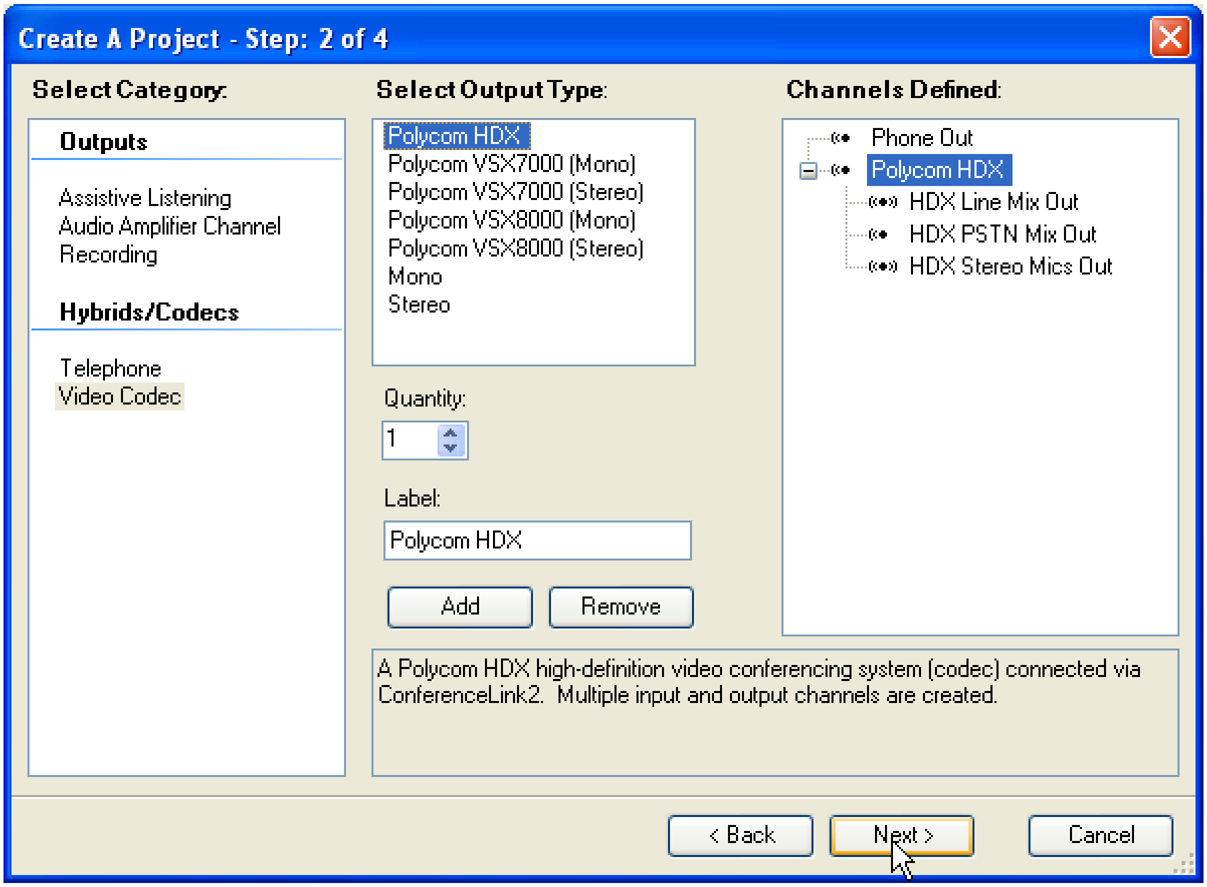
The Codec Program Audio In and Codec Video Call In channels have dynamics processing, parametric equalization, an input fader, input delay, and mute control available for their input processing. In addition there are signal level meters that can be displayed for these channels.

The Codec Voice Call In and Codec UI Audio In channels are routed directly to the SoundStructure matrix and do not have dedicated SoundStructure input processing or signal level metering. If processing or metering is desired on these signals before the signals are used in the matrix, these signals may be routed to the SoundStructure submixes where dynamics processing, parametric equalization, fader, delay mute control, and signal level meters are available. The outputs from the submixes may then be used as inputs to the matrix. As with other virtual channels, the submix signals have virtual channel names and are controlled in the same fashion as any other virtual channel within a SoundStructure system.

Understanding The Polycom Video Codec Output Channels

SoundStructure Studio creates several output virtual channels that are sent to the Polycom Video Codec system as shown in the following figure.

Polycom Video Codec Output Virtual Channels


Create A Project - Step: 2 of 4
Select Category:
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Telephone
Video Codec
Select Output Type:
Polycom HDX
Polycom VSX7000 (Mono)
Polycom VSX7000 (Stereo)
Polycom VSX8000 (Mono)
Polycom VSX8000 (Stereo)
Mono
Stereo
Add
Remove
Channels Defined:
Phone Out
Polycom HDX
HDX Line Mix Out
HDX PSTN Mix Out
HDX Stereo Mics Out
< Back
Next >
Cancel

Quantity: 1

Label: Polycom HDX

A Polycom HDX high-definition video conferencing system (codec) connected via ConferenceLink2. Multiple input and output channels are created.

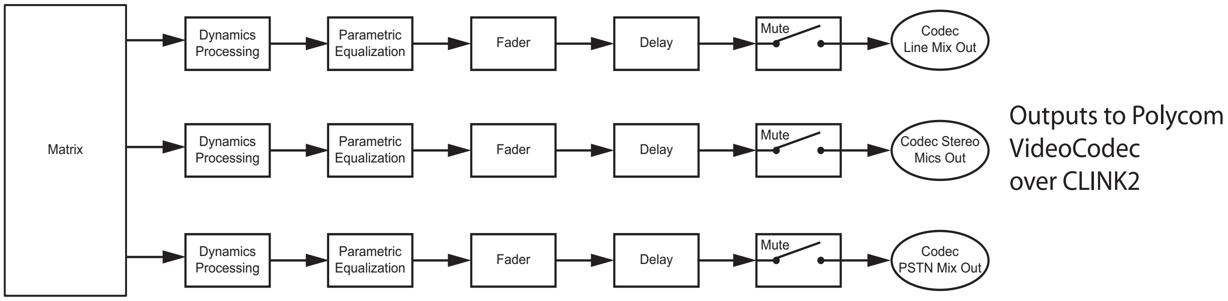
The output channels sent to the Polycom Video Codec are described in the following table.

Polycom Video Codec Output Channels

Signal from SoundStructure	*Description*
Codec Line Out Mix	This is a stereo virtual channel that is sent to all outgoing call mixes on the Polycom Video Codec and to the VCR/DVD output connections.
Codec Voice Call Mix Out	A mono virtual channel that contains a mix of the telephony receive signals from any telephony plug-in cards on the SoundStructure system.
Codec Stereo Mics Out	A stereo virtual channel that is routed to the remote video participants of the Polycom Video Codec conferencing system and to the VCR/DVD output on the Polycom Video Codec.

The output processing on SoundStructure that is available for these output channels is shown in the following figure. All signals have the same processing that includes dynamics, parametric equalization, fader, delay, and mute. All the signals that are sent to the Polycom Video Codec system have signal level meters that are displayed on the Channels page.

. Polycom Video Codec Output Processing Channels


Matrix
Dynamics Processing
Parametric Equalization
Fader
Delay
Mute
Codec Line Mix Out
Codec Stereo Mics Out
Codec PSTN Mix Out
Outputs to Polycom VideoCodec over CLINK2

Routing The Polycom Video Codec Signals

The Polycom Video Codec system receives the SoundStructure output signals and internal to the Video Codec mixes the signals it needs to create the transmit signals to the Codec Voice Call interface and Codec Video interface. These signals are mixed as follows:

The transmit signal to the remote video participants is mixed within the Polycom Video Codec to include:

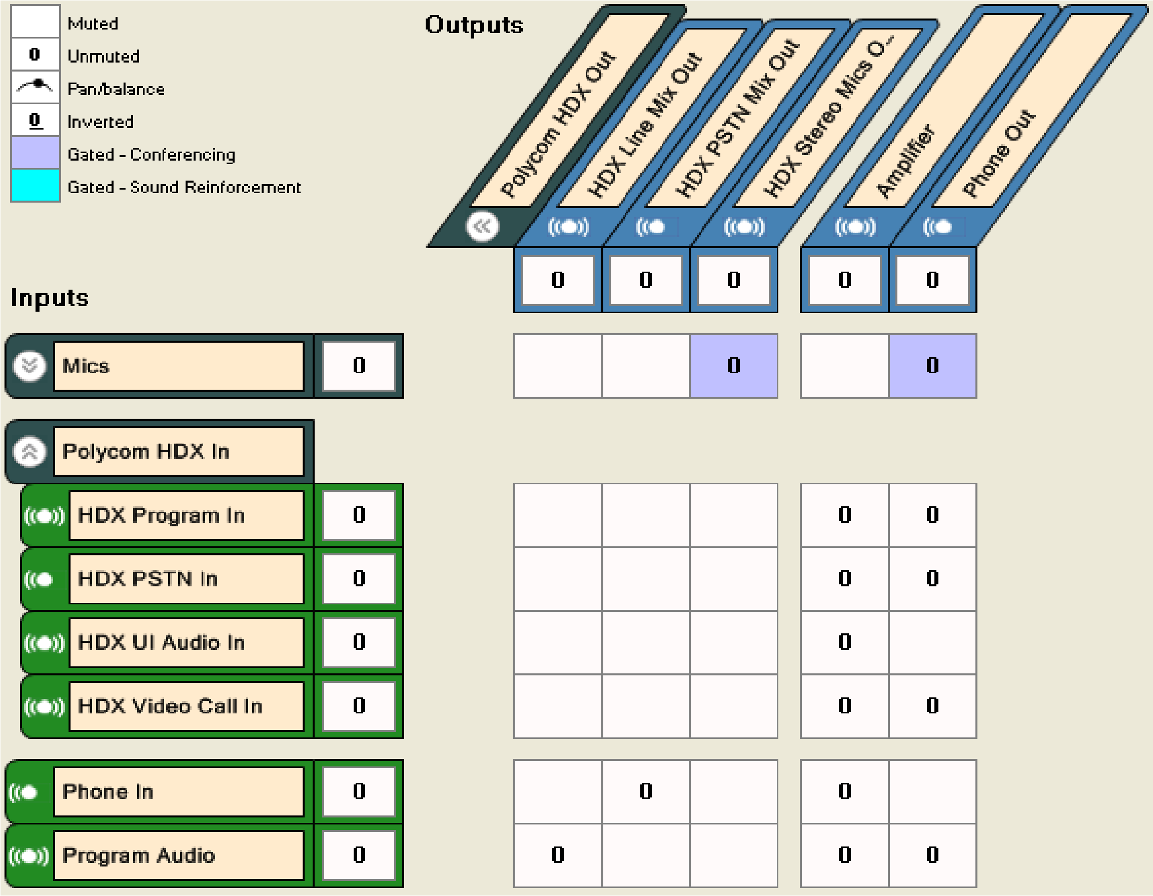
- Codec Voice Call Mix Out
- Codec Stereo Mics Out
- Codec Line Mix Out

The transmit signal to the remote telephony (PSTN) Video Codec participants includes the remote video participant audio and:

- Codec Voice Call Mix Out
- Codec Stereo Mics Out
- Codec Line Mix Out

This default routing inside the Polycom Video Codec means that the SoundStructure matrix does not have to add these channels to the Codec Stereo Mics Out signal. Typically the SoundStructure matrix looks like the following figure where the SoundStructure “Phone In” signal is routed to the “Codec Voice Call Mix Out” channel, the SoundStructure “Program Audio” signal is routed to the “Codec Line Mix Out” channel, and the SoundStructure “Mics” group is routed to the “Codec Stereo Mics Out” channel.

Matrix Channel for the Polycom Video Codec in SoundStructure Studio


Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Polycom HDX Out
Inputs
Polycom HDX In

Inputs		HDX Line Mix Out	HDX PSTN Mix Out	HDX Stereo Mics O...	Amplifier	Phone Out
		0	0	0	0	0
Mics	0			0		0
HDX Program In	0				0	0
HDX PSTN In	0				0	0
HDX UI Audio In	0				0	
HDX Video Call In	0				0	0
Phone In	0		0		0	
Program Audio	0	0			0	0

Using the Mute Controls

In firmware earlier than v1.3 SoundStructure firmware, any change in the mute state of the Polycom table and ceiling microphones causes the SoundStructure device to receive either commands depending on whether the Video Codec system is being muted or unmuted.

```
set mute “Mics” 1
set mute “Mics” 0
```

No audio paths are muted inside the Polycom Video Codec when a Video Codec, that is connected to a SoundStructure device over CLink2 interface, receives a mute command. The only effect of the Video Codec receiving a mute command is that the SoundStructure device is sent a mute message as described above. It is required that the SoundStructure device perform the muting.

Note: Muting SoundStructure Microphones

In pre-1.3 SoundStructure firmware, if the SoundStructure system's microphones are muted independently of the Polycom Video Codec system, the Polycom Video Codec mute status may not reflect the actual SoundStructure mute status.

In 1.3 SoundStructure firmware and later, SoundStructure Events may be used to link the SoundStructure mute to the Video Codec mute and vice versa.

Note: Muting Audio Not Supported in CLink2 Interface

No audio paths are muted internal to the Polycom Video Codec system when a mute command is sent to an Video Codec system that is connected to a SoundStructure device over the CLink2 interface. The muting must occur within the SoundStructure device.

Any mute command sent to the Video Codec triggers the mute command shown above which causes all the signals on the SoundStructure device that are members of the “Mics” virtual channel group to be muted or unmuted, respectively. By default the “Mics” virtual channel group is created by SoundStructure Studio and includes all the local microphone virtual channels. A SoundStructure command status message is sent out to the SoundStructure control ports indicating the mute status has changed.

Muting the SoundStructure microphones does not affect the routing of an attached PSTN telephone caller on the SoundStructure to the remote Video Codec participants. In other words, by default the local SoundStructure participants is muted to all remote participants while the remote telephony participants and remote video participants are able to talk to each other.

Using Advanced Muting Applications

By default, a SoundStructure design automatically defines the “Mics” virtual channel group and places all the microphones in the design in that group. The membership of this group may be changed and other signals placed into the “Mics” group if it is desired to change the behavior of how the mute command from the Video Codec maps to the audio signals within a SoundStructure system. It is possible to put line level input sources (such as program audio) or even the output signal that is sent to the Video Codec into the “Mics” group and have those signals be affected when the Polycom Video Codec mute status is changed. Although the name of the group is “Mics”, any virtual channel can be part of the group.

As another example, it is possible to rename the current “Mics” virtual channel group to another name and create a submix called “Mics” and have that virtual channel be muted instead of the default “Mics” group. This could be used to allow in-room reinforcement, for example, while the “Mics” submix would be muted to prevent that audio from being transmitted to the remote participants.

There is tremendous design flexibility by mapping the Video Codec Mute command to affect the “Mics” virtual channel or virtual channel group. If there is no “Mics” virtual channel or virtual channel group defined, then no audio paths are muted when the end user mutes the Polycom Video Codec system directly.

Note: Muting Audio Not Supported When Muted on Video Codec

In pre-1.3 SoundStructure firmware, If the “Mics” definition is not present on the SoundStructure device, no audio path is muted when the user mutes the Video Codec. It is the system integrators responsibility to ensure that the Video Codec mute signal is mapped effectively to the SoundStructure if the definition of the “Mics” virtual channel group is changed.

Due to the flexibility of SoundStructure events in SoundStructure firmware version 1.3 or later, a name other than “Mics” may be used and the system operates properly.

Using the Volume Controls

The volume setting of a Polycom Video Codec system is sent automatically to the SoundStructure device via the Conference Link2 interface whenever the volume changes on the Polycom Video Codec system. In pre-1.3 SoundStructure firmware, if the volume changes on the SoundStructure system, the Polycom Video Codec does ***not*** receive the volume change event from the SoundStructure device. Only when volume change commands are sent to the Polycom Video Codec via a control system or infrared remote the volume event is automatically transmitted to the SoundStructure device.

In the SoundStructure devices the volume value from the Polycom Video Codec is mapped to the output fader control on the SoundStructure virtual channel called “Amplifier”. The mapping subtracts 30 from the Video Codec volume setting to create the level to be set on the output fader. The Video Codec volume settings can range from 0 to 50 which maps to the SoundStructure fader range of -30 to +20.

The fader command executed on the SoundStructure device is:

```
set fader “Amplifier” x
```

where x is the Video Codec volume level minus 30.

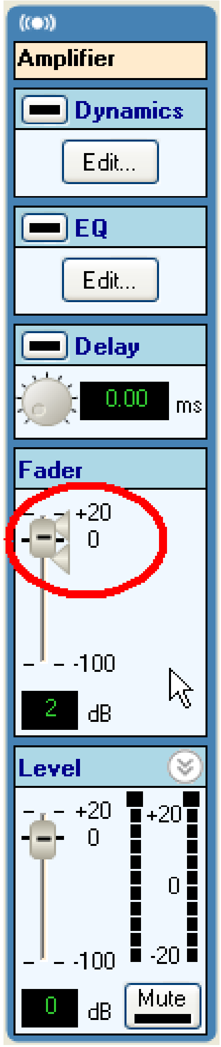
At the maximum volume setting for the Polycom Video Codec (50), this causes the SoundStructure to execute the command

```
set fader “Amplifier” 20
```

This sets the fader control for the virtual channel “Amplifier” to 20.

A command status message is sent out to the SoundStructure control ports indicating the new fader level.

It is also possible to limit the minimum and maximum user gain settings via SoundStructure Studio software by using the min and max gain limits on the fader control. This can be done graphically on the channels page as shown in the following figure or via the SoundStructure API. See the fader command for the syntax of how to use the min and max user limits.



Because SoundStructure systems receive volume change requests from the Video Codec, and the pre-1.3 SoundStructure firmware does not send volume messages to the Polycom Video Codec, any volume limit set on the SoundStructure system is not recognized by the Video Codec. This means that while the user adjusts volume on the Polycom Video Codec, the request does not display as the volume continues to change on the Polycom Video Codec UI although a volume limit may have already been reached within the SoundStructure system which would prevent the system from getting any louder in the room.

SoundStructure devices with version 1.3 firmware or later use events to both receive and set the Video Codec volume parameter. See Using Events, Logic, and IR for information on using events with an Video Codec.

Designing With Polycom Digital Microphone Arrays

Each Polycom digital microphone array has three microphone elements and must be thought of as *three microphone inputs*. As a result, each Polycom digital microphone requires the processing of three SoundStructure analog input channels. In other words, for Because each digital microphone array is represented as three microphones, every microphone array and its respective three microphone elements can be used independently with a SoundStructure device. This means that several Polycom microphone arrays can be linked together and used, for example, in room combining applications where one or more microphone arrays are in one room and one or more microphone arrays are in a different room. The different microphone array elements may be muted and used in the matrix independently as easily as if they were traditional analog microphones.

Note: Representing Microphones on SoundStructure

Each digital microphone is represented as three microphones on a SoundStructure device.

As shown in the following figure, the three microphone elements are labeled as A, B, and C within SoundStructure Studio software environment. The ceiling microphone arrays have an orientation dot on the band that indicates element A. The orientation of the microphone array is only significant in stereo or positional conferencing applications where it is important to have the relative position of microphone elements with respect to the video conferencing camera. See Creating Advanced Applicationsfor examples of stereo video conferencing applications.

Labeling Microphone Elements

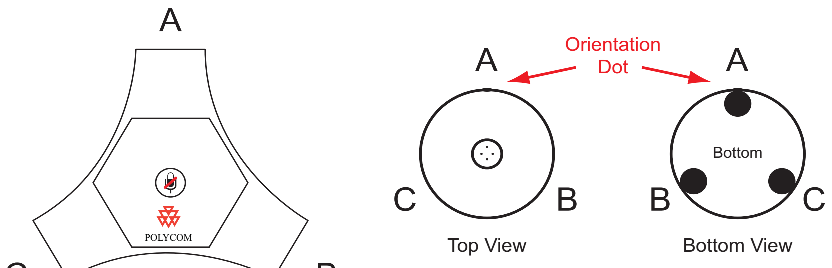


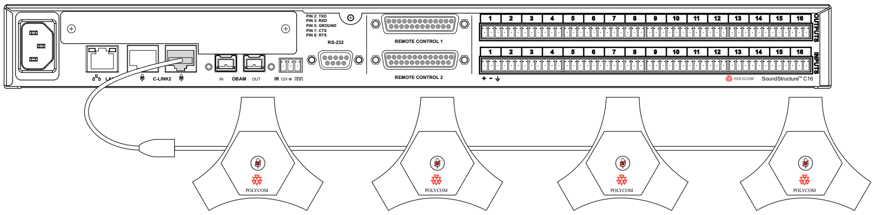
Table Mic Array

Ceiling Mic Array

Understanding Digital Microphone Cabling Requirements

Up to four microphone arrays may be used with the SoundStructure products depending on the particular SoundStructure model as described in the following figure.

Microphone Arrays Connected to SoundStructure Devices



The following table shows the number of analog inputs that are available based on the number of microphone arrays that are used in a system. As an example, a SoundStructure C16 supports 16 analog inputs. When used with two microphone arrays, 10 analog inputs are still available for use with other analog inputs including microphones, program audio, etc.

Analog Inputs Available for Connected Microphone Arrays

		Number of Available Analog Inputs		
		C16	C12	C8
Number of Polycom Microphones	**0**	16	12	8
	1	13	9	5
	2	10	6	2
	3	7	3	--
	4	4	0	--

Note: Connecting Digital Microphone Arrays to CLink2

In SoundStructure only applications, connect the digital array microphones to the ***right*** CLink2 port (the port closest to the OBAM interface).

In SoundStructure and Video Codec applications, connect the Video Codec to the ***left*** CLink2 port on SoundStructure and connect digital microphones to either CLink2 port on the Video Codec system.

Version 2.0.1 of Video Codec only supports 3 microphone arrays when connected to the SoundStructure device over CLink2. Later version of Video Codec firmware support up to 4 microphone arrays on a SoundStructure device.

Updating Digital Microphone Firmware

When the digital microphone arrays are connected directly to the SoundStructure device, the version of firmware on the microphones are compared to the version of microphone firmware included within the SoundStructure device. If the version of firmware on the microphones is older than the version of firmware included with the SoundStructure firmware, the microphones are automatically updated with the version firmware from SoundStructure.

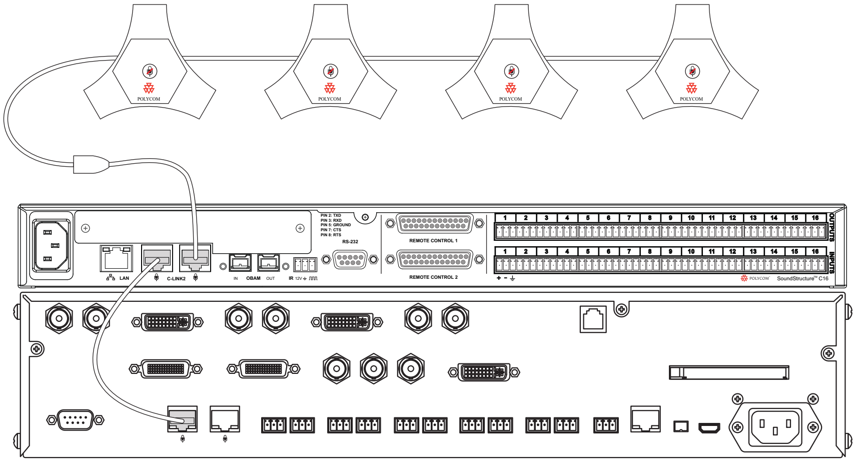
Version 26 of the microphone firmware is required for operation with SoundStructure devices. Microphones that are plugged directly into the right CLink2 port on a SoundStructure device (assuming SoundStructure firmware version 1.1.2 is used) is updated to version 26 if it is necessary to update the microphone arrays. Once updated, the microphones continue to use version 26 even if they are unplugged or powered down.

Note: Required Microphone Array Version

Version 26 or later of the microphone array is required for operation with the SoundStructure devices.

The SoundStructure device logs may be viewed to show the number and version of microphones connected. An example of the log is shown in the next figure. The first entry is shown when devices are plugged into the SoundStructure. In this example the SoundStructure is connected to a Video Codec via its left Clink2 port and then there are 4 microphones connected to the right Clink2 port as shown in the following figure.

SoundStructure Device Log Showing Connected Microphones



Below is an excerpt from the SoundStructure log file:

```
Feb  8 23:16:40 soundstructure cmdd: sts: conference link configured
Feb  8 23:16:40 soundstructure cmdd: sts:    [0] Video Codec
Feb  8 23:16:40 soundstructure cmdd: sts:    [1] SoundStructure
Feb  8 23:16:40 soundstructure cmdd: sts:    [2] Polycom Mic (f/w 26)
Feb  8 23:16:40 soundstructure cmdd: sts:    [3] Polycom Mic (f/w 26)
Feb  8 23:16:40 soundstructure cmdd: sts:    [4] Polycom Mic (f/w 26)
Feb  8 23:16:40 soundstructure cmdd: sts:    [5] Polycom Mic (f/w 26)
```

According to the log all microphones have version 24 firmware installed.

Because microphone arrays may be shipped with a firmware version that may be earlier than version 26, the firmware should be updated once to revision 26 by connecting the microphones directly to the right CLink2 port (the port closest to the OBAM interface) on SoundStructure device for 30 seconds.

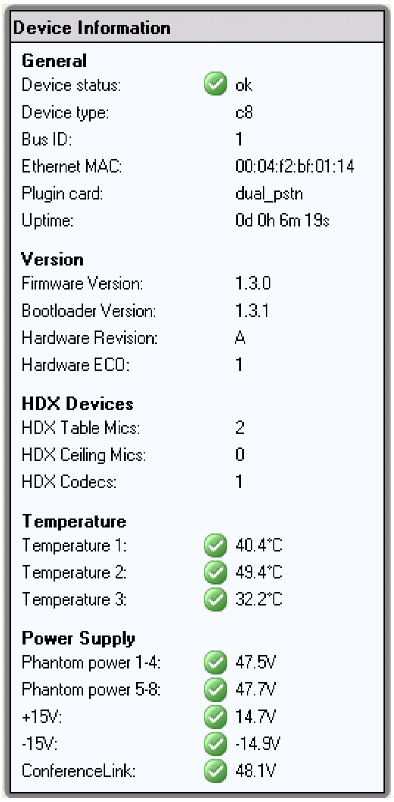
Note: Microphone Firmware Compatibility

To make sure the firmware on the microphone arrays is compatible with the SoundStructure device, during the installation process plug the microphone chain (up to four microphones may be cascaded during this process) into the right CLink2 port of SoundStructure for 30 seconds to ensure the firmware is updated to the version required for SoundStructure operation.

This process only needs to be done once, even if the microphones ultimately are connected directly into the Video Codec and not the SoundStructure device.

Detecting CLink2 Devices

When connected to a SoundStructure device, the wiring page shows the status of the number and type of CLink2 devices. This information is shown in the following figure where two table mics and one Polycom Video Codec were discovered. To have this information automatically updated as devices are connected over CLink2, select the poll device information check box on the top of the wiring page.



Device Information

General		
Device status:	✓	ok
Device type:		c8
Bus ID:		1
Ethernet MAC:		00:04:f2:bf:01:14
Plugin card:		dual_pstn
Uptime:		0d 0h 6m 19s
Version		
Firmware Version:		1.3.0
Bootloader Version:		1.3.1
Hardware Revision:		A
Hardware ECO:		1
HDX Devices		
HDX Table Mics:		2
HDX Ceiling Mics:		0
HDX Codecs:		1
Temperature		
Temperature 1:	✓	40.4°C
Temperature 2:	✓	49.4°C
Temperature 3:	✓	32.2°C
Power Supply		
Phantom power 1-4:	✓	47.5V
Phantom power 5-8:	✓	47.7V
+15V:	✓	14.7V
-15V:	✓	-14.9V
ConferenceLink:	✓	48.1V

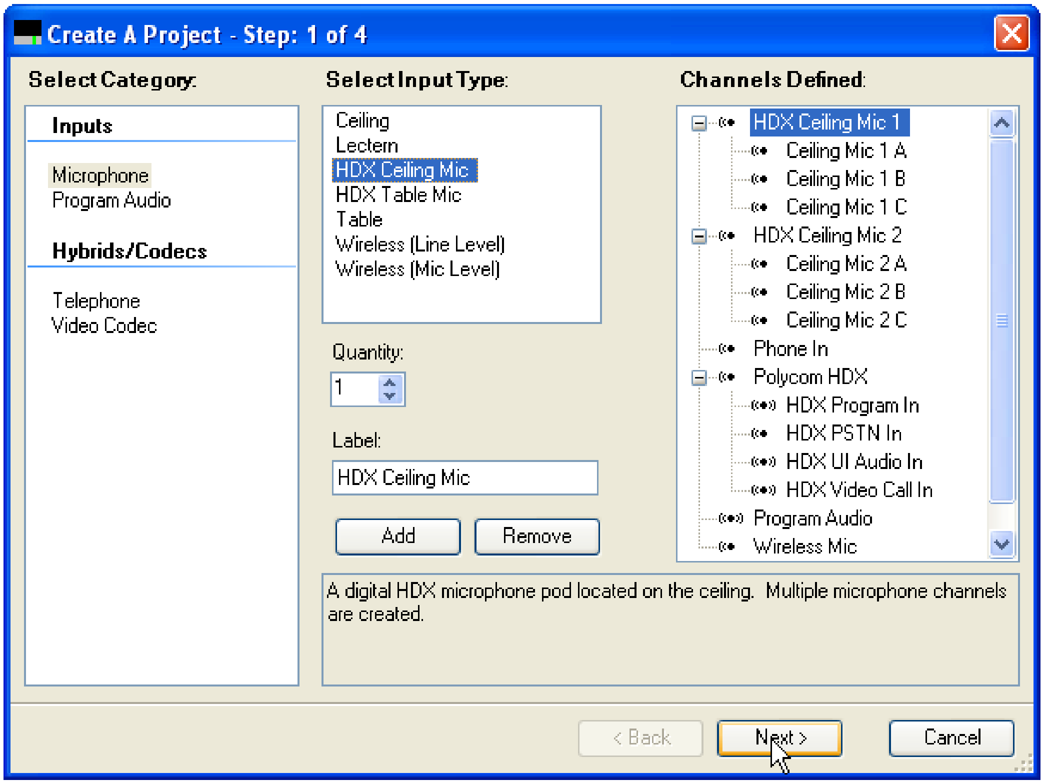
Status and Type of Conference Link2 Devices on the Wiring Page

Viewing Digital Microphone Array Example

As an example of using the digital microphone arrays, consider a design that uses two ceiling microphone arrays, one wireless analog microphone, a stereo program audio source, a Polycom Video Codec conferencing system, a telephone line, and a stereo amplifier.

The first step of the design process is to select the input signals as shown in the following figure. Notice that for each Polycom ceiling microphone array that is added, there are three mono microphones with names that include A, B, and C that are added to the project.

Selecting Input Signals for Microphone Arrays in SoundStructure Studio


Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Telephone
Video Codec
Select Input Type:
Ceiling
Lectern
HDX Ceiling Mic
HDX Table Mic
Table
Wireless (Line Level)
Wireless (Mic Level)
Channels Defined:
HDX Ceiling Mic 1
Ceiling Mic 1 A
Ceiling Mic 1 B
Ceiling Mic 1 C
HDX Ceiling Mic 2
Ceiling Mic 2 A
Ceiling Mic 2 B
Ceiling Mic 2 C
Phone In
Polycom HDX
HDX Program In
HDX PSTN In
HDX UI Audio In
HDX Video Call In
Program Audio
Wireless Mic
Add
Remove
< Back
Next >
Cancel

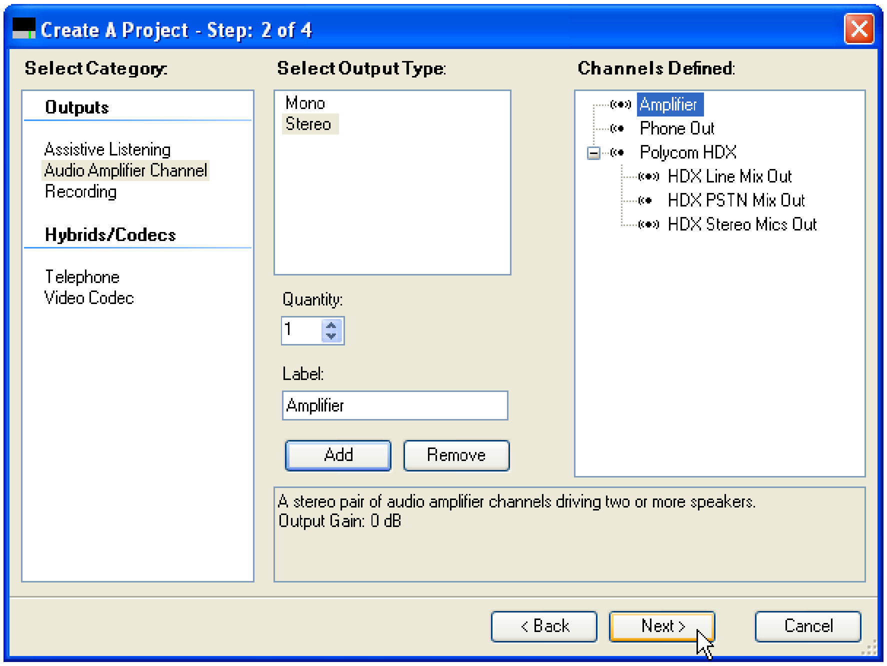
Quantity: 1

Label: HDX Ceiling Mic

A digital HDX microphone pod located on the ceiling. Multiple microphone channels are created.

The second step of the design process is to select the outputs from the system as shown in the following figure.

Selecting Outputs for Microphone Arrays in SoundStructure Studio


Create A Project - Step: 2 of 4
Select Category:
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Telephone
Video Codec
Select Output Type:
Mono
Stereo
Channels Defined:
Amplifier
Phone Out
Polycom HDX
HDX Line Mix Out
HDX PSTN Mix Out
HDX Stereo Mics Out
Add
Remove
< Back
Next >
Cancel

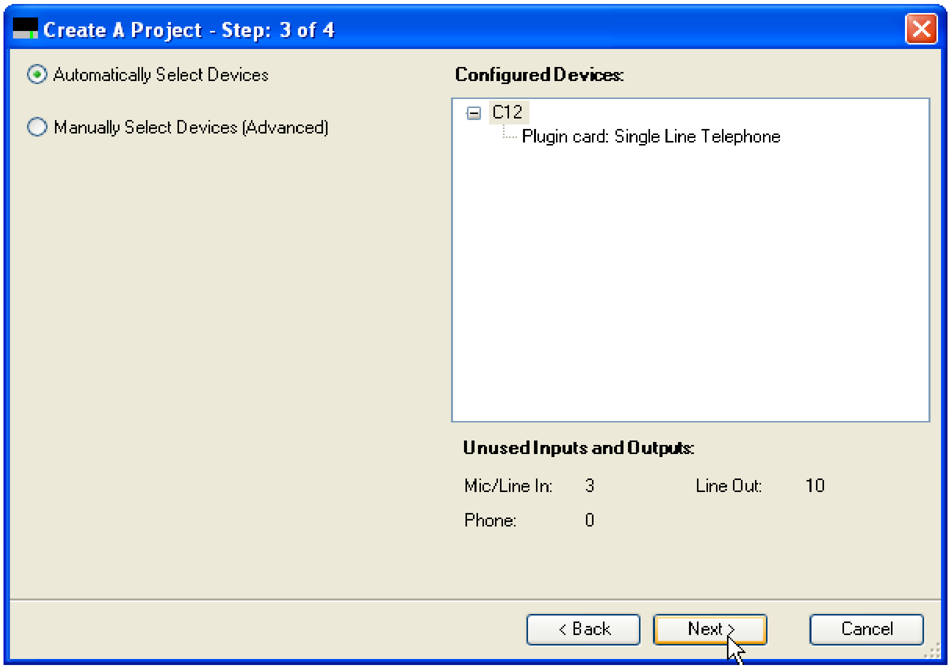
Quantity: 1

Label: Amplifier

A stereo pair of audio amplifier channels driving two or more speakers.
Output Gain: 0 dB

In the third step, the equipment is selected. In this case a C12 is required and has three additional analog inputs available to use after the system is designed.

Selecting Equipment for Microphone Arrays in SoundStructure Studio


Create A Project - Step: 3 of 4
Configured Devices:
< Back
Next >
Cancel

● Automatically Select Devices

○ Manually Select Devices (Advanced)

C12
- Plugin card: Single Line Telephone

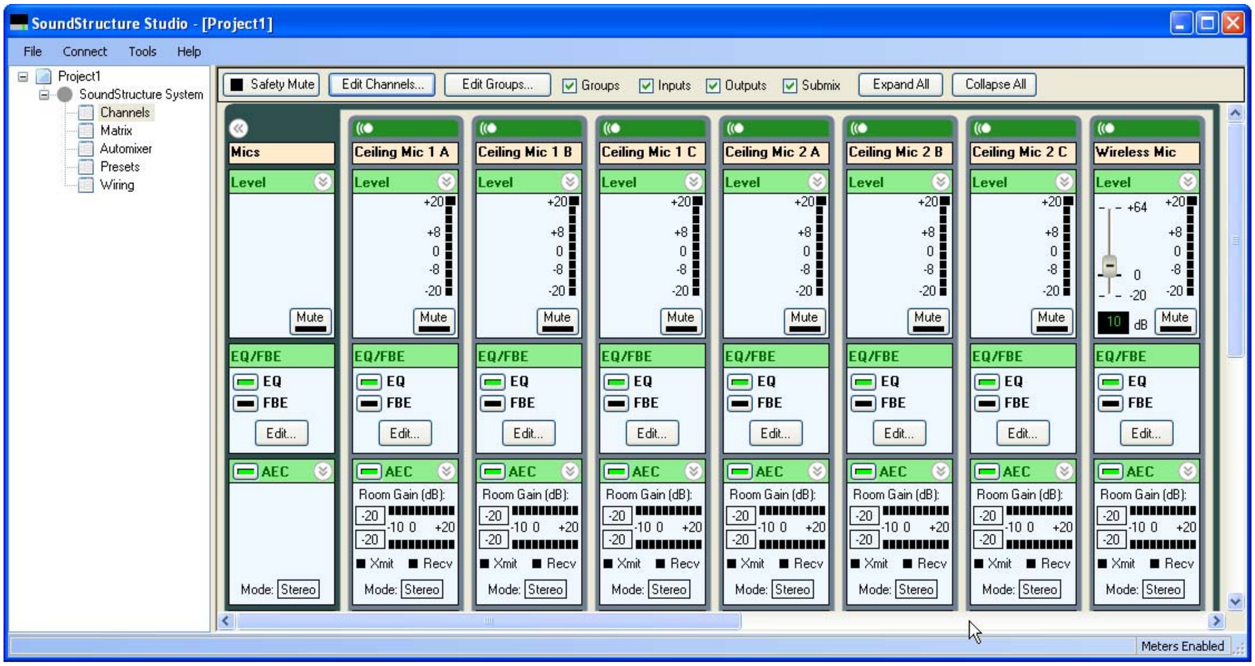
Unused Inputs and Outputs:

Mic/Line In:	3	Line Out:	10
Phone:	0		

In the final step, offline operation is selected and the resulting project is created. The channels page associated with this project is shown in the following figure.

Notice that the digital microphone arrays are shown just as any other audio channel in the system. Although the signal level meters are active for the Polycom microphones, one difference between an analog microphone input and a digital microphone array input to SoundStructure is that the analog gain slider is not present on the digital microphones as shown in the following figure. The slider is not present because it does not affect the signal level coming in from the microphone array as the signal is already digitized in the microphone array and does not pass through the SoundStructure's analog gain stage.

Viewing Digital Microphone Arrays


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Expand All
Collapse All
Mics
Ceiling Mic 1 A
Ceiling Mic 1 B
Ceiling Mic 1 C
Ceiling Mic 2 A
Ceiling Mic 2 B
Ceiling Mic 2 C
Wireless Mic
Level
+64
+20
+8
0
-8
-20
10 dB
Mute
EQ/FBE
EQ
FBE
Edit...
AEC
Room Gain (dB):
-20
-10 0 +20
Xmit
Recv
Mode: Stereo
Meters Enabled

Assigning Digital Microphone Array Channels To Physical Inputs

When Polycom digital microphone arrays are used within SoundStructure Studio, SoundStructure Studio assigns the processing for each digital microphone input from a physical analog input. SoundStructure Studio reserves processing by starting with the last analog input channel and working towards the first analog input.

For example, if a single Polycom digital microphone array is used with a SoundStructure C12, the processing from physical analog inputs 12, 11, and 10 are used for the digital microphone elements A, B, and C respectively and the physical inputs 12, 11, and 10 are not able to be used for any analog inputs. If two digital microphones are used with a C12, the second digital microphone's elements A, B, and C utilize the processing associated with analog physical inputs 9, 8, and 7 respectively. In this example, analog input signals may not be connected to inputs 7-12.

Note: Processing Paths for Polycom Digital Microphone Array

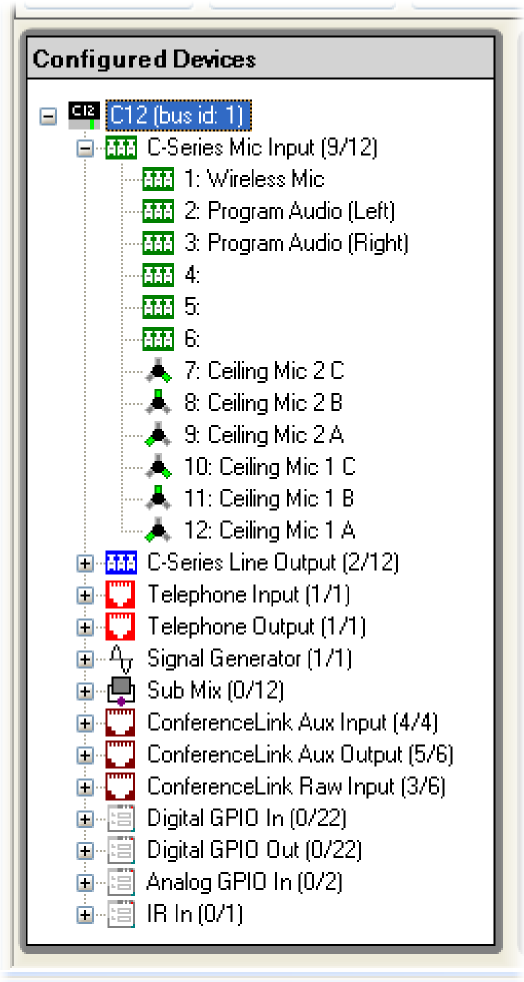
Using Polycom digital microphone array inputs requires the same processing paths that are used with analog input signals. When Polycom digital microphones are used, any analog signals on the physical inputs assigned to the Polycom microphone elements are not used.

When analog inputs are also used as part of the design, such as for a wireless microphone and program audio in this example, the analog inputs that are used are allocated from the first analog input to the last available input.

This allocation and assignment of analog inputs can be viewed from the wiring page as shown in the following figure. Note that the particular microphone element associated with the labeling A, B, and C is highlighted in green on the wiring page for each digital microphone input.

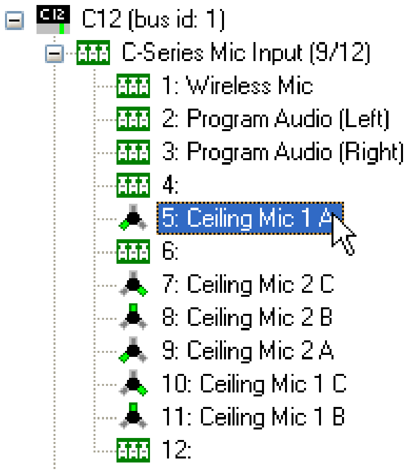
In this example, up to six analog inputs can be used (three analog inputs are presently in use) in addition to the two Polycom ceiling microphones.

Analog Inputs and Polycom Ceiling Microphones on Wiring Page


Configured Devices
C12 (bus id: 1)
C-Series Mic Input (9/12)
1: Wireless Mic
2: Program Audio (Left)
3: Program Audio (Right)
4:
5:
6:
7: Ceiling Mic 2 C
8: Ceiling Mic 2 B
9: Ceiling Mic 2 A
10: Ceiling Mic 1 C
11: Ceiling Mic 1 B
12: Ceiling Mic 1 A
C-Series Line Output (2/12)
Telephone Input (1/1)
Telephone Output (1/1)
Signal Generator (1/1)
Sub Mix (0/12)
ConferenceLink Aux Input (4/4)
ConferenceLink Aux Output (5/6)
ConferenceLink Raw Input (3/6)
Digital GPIO In (0/22)
Digital GPIO Out (0/22)
Analog GPIO In (0/2)
IR In (0/1)

The digital microphone array elements can be moved on the wiring page to different physical inputs if desired by clicking and dragging the microphone to move it to a different physical input. The following figure shows moving Ceiling Mic 1 A from input 12 to input 5 to make it possible to connect an analog input to input 12.

Moving Microphones between Physical Inputs



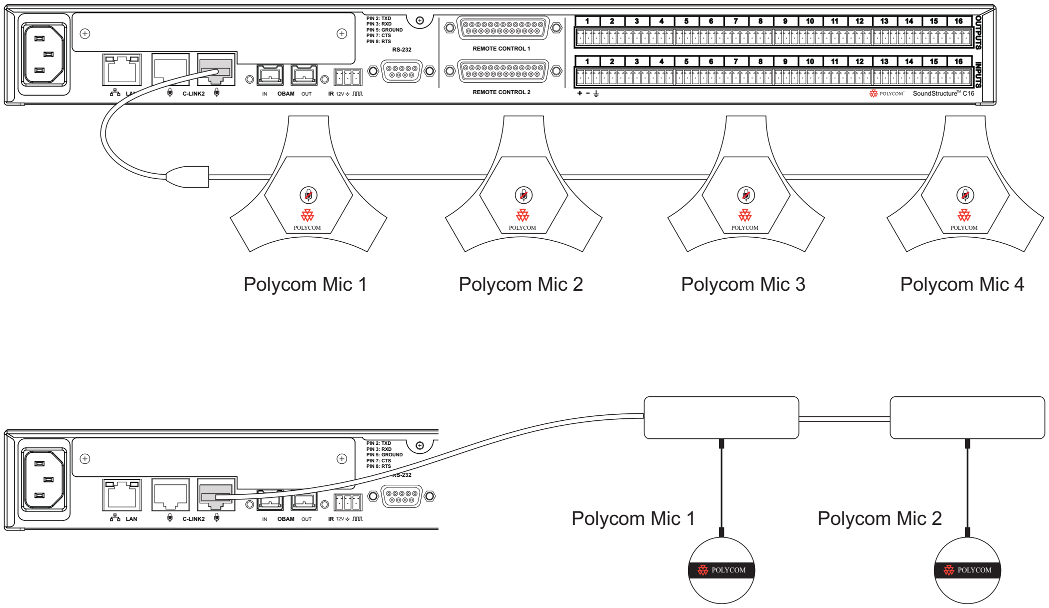
C12 (bus id: 1)
C-Series Mic Input (9/12)
1: Wireless Mic
2: Program Audio (Left)
3: Program Audio (Right)
4:
5: Ceiling Mic 1 A
6:
7: Ceiling Mic 2 C
8: Ceiling Mic 2 B
9: Ceiling Mic 2 A
10: Ceiling Mic 1 C
11: Ceiling Mic 1 B
12:

If any changes are made to the wiring page, please make sure to save the file from the File Save menu option so that the updated virtual channel definitions are saved in the configuration.

Numbering Digital Microphone Array

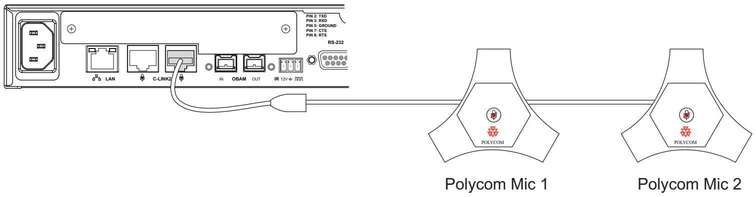
Examples of the microphone connections and their numbering within SoundStructure are shown in the following figure.

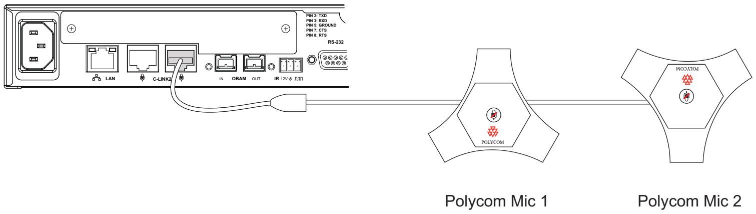
Microphone Connections within SoundStructure

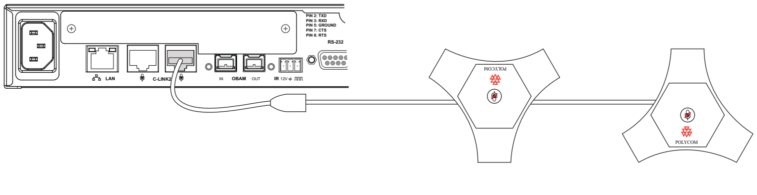

LAN
C-LINK2
IN
OBAM
OUT
IR
RS-232
REMOTE CONTROL 1
REMOTE CONTROL 2
OUTPUTS
INPUTS
SoundStructure C16
Polycom Mic 1
Polycom Mic 2
Polycom Mic 3
Polycom Mic 4
Polycom Mic 1
Polycom Mic 2

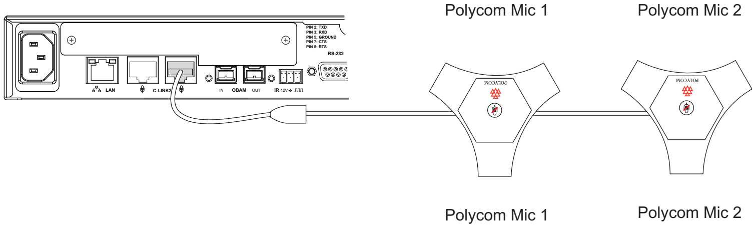
The orientation of the microphone does not affect the sequential numbering as shown in the following figure.

Sequential Numbering of Polycom Microphones


PIN 2: TXD
PIN 3: RXD
PIN 5: GROUND
PIN 7: CTS
PIN 8: RTS
RS-232
LAN
C-LINK2
IN
OBAM
OUT
IR
POLYCOM
Polycom Mic 1
Polycom Mic 2

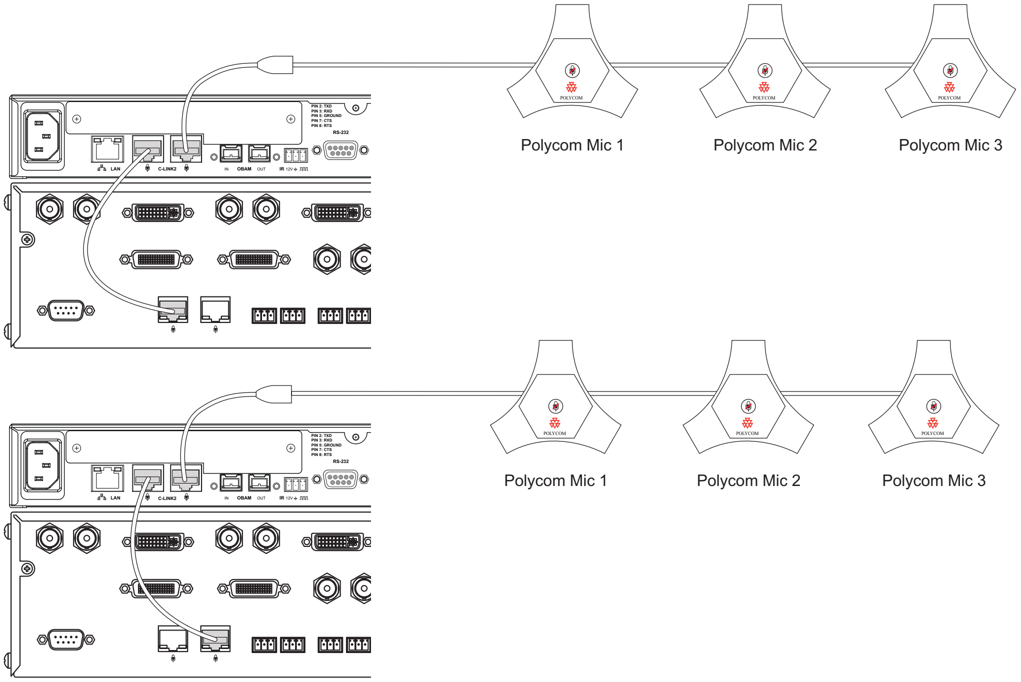

PIN 2: TXD
PIN 3: RXD
PIN 5: GROUND
PIN 7: CTS
PIN 8: RTS
RS-232
LAN
C-LINK2
IN
OBAM
OUT
IR
POLYCOM
Polycom Mic 1
Polycom Mic 2


PIN 2: TXD
PIN 3: RXD
PIN 5: GROUND
PIN 7: CTS
PIN 8: RTS
RS-232
LAN
C-LINK2
IN
OBAM
OUT
IR
POLYCOM
Polycom Mic 1
Polycom Mic 2


PIN 2: TXD
PIN 3: RXD
PIN 5: GROUND
PIN 7: CTS
PIN 8: RTS
RS-232
LAN
C-LINK2
IN
OBAM
OUT
IR
POLYCOM
Polycom Mic 1
Polycom Mic 2

When a Polycom Video Codec system is also connected over the CLink2 interface and the digital microphones connected directly to the SoundStructure device, the numbering of the digital microphone arrays are the same as the previous figures.

Numbering of Digital Microphone Arrays with Conference Link2 Devices

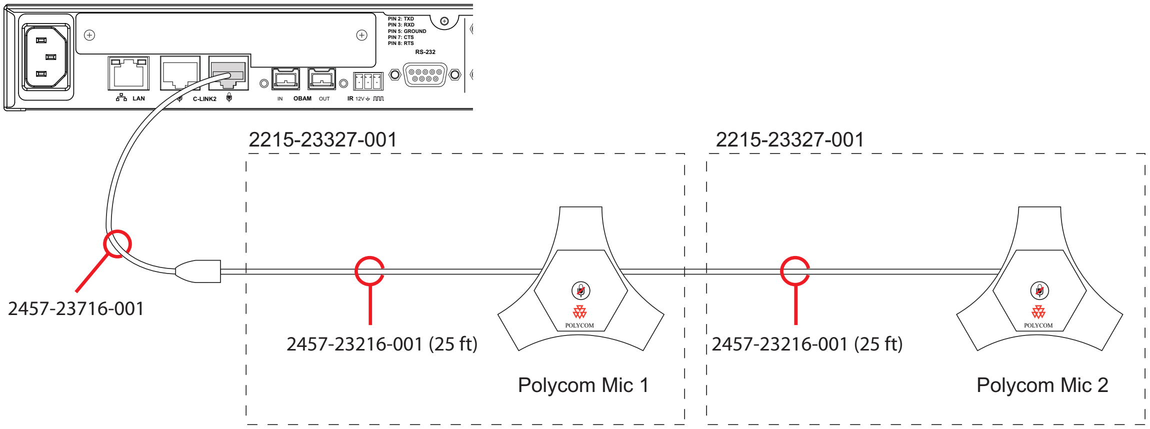

Polycom Mic 1
Polycom Mic 2
Polycom Mic 3
LAN
C-LINK2
IN
OBAM
OUT
IR
RS-232
Polycom Mic 1
Polycom Mic 2
Polycom Mic 3
LAN
C-LINK2
IN
OBAM
OUT
IR
RS-232

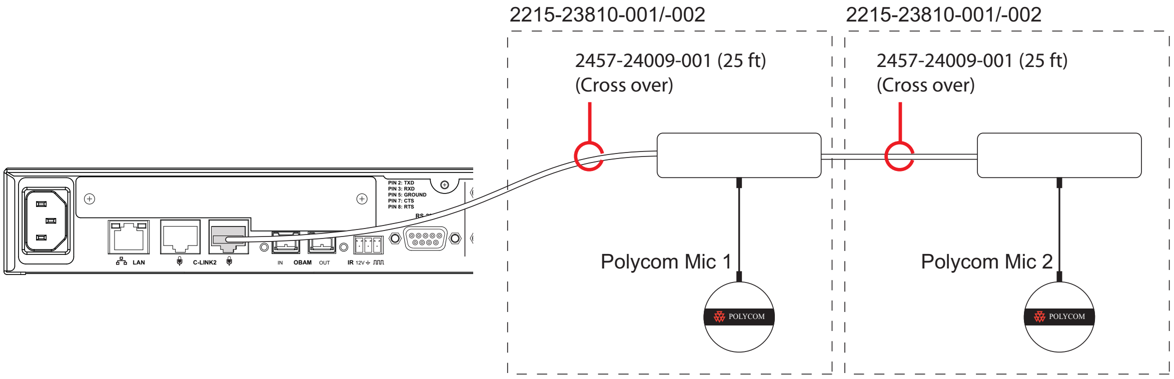
Understanding Installation Options

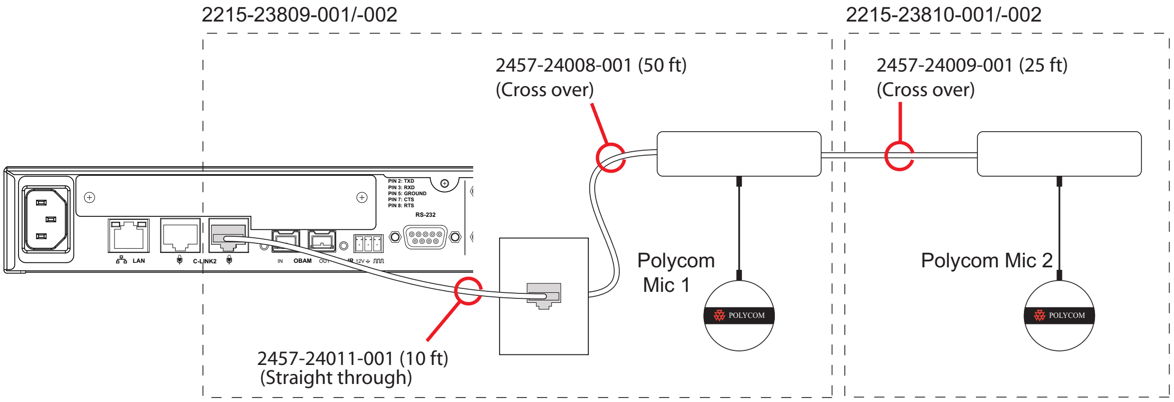
There are several installation options available depending on whether tabletop or ceiling microphones are being used.

The following figure shows typical wiring options using the Polycom SKUs highlighted with the dashed boxes for tabletop microphones and ceiling microphone arrays.

Wiring Options with Polycom SKUs for Polycom Tabletop Microphones


2215-23327-001
2215-23327-001
2457-23716-001
2457-23216-001 (25 ft)
2457-23216-001 (25 ft)
Polycom Mic 1
Polycom Mic 2


2215-23810-001/-002
2215-23810-001/-002
2457-24009-001 (25 ft)
(Cross over)
2457-24009-001 (25 ft)
(Cross over)
Polycom Mic 1
Polycom Mic 2


2215-23809-001/-002
2215-23810-001/-002
2457-24008-001 (50 ft)
(Cross over)
2457-24009-001 (25 ft)
(Cross over)
Polycom
Mic 1
Polycom Mic 2
2457-24011-001 (10 ft)
(Straight through)

These SKU’s include the cables that are shown within the dashed boxes and are summarized in the table below.

See the Polycom SoundStructure and HDX Microphones Best Practices Guide for more information on using Polycom microphones with SoundStructure devices.

SKUs and Cables for Polycom Tabletop Microphones

SKU	Description
2215-23327-001	Tabletop microphone array with 25’ Walta to Walta cable
2215-23809-001	Black ceiling microphone array kit
2215-23809-002	White ceiling microphone array kit
2215-23810-001	Black ceiling microphone array ***extension*** kit
2215-23810-002	White ceiling microphone array ***extension*** kit

For reference, the Walta connector is the flat connector that is on the side of the tabletop microphone arrays and the RJ45 connector is compatible with the connectors on the rear of the SoundStructure device and on the digital ceiling microphone array.

The digital tabletop microphone arrays are connected via Walta terminated cables and then the last cable is terminated into the SoundStructure via the Walta to RJ45 interface cable.

The digital ceiling microphone arrays are connected via RJ45 terminated cables and may be connected directly to the rear-panel of the SoundStructure.

The maximum length of all the conference link cables should not exceed 175 ft and no single run of cable should exceed 100 ft.

Note: Maximum Length of Conference Link Cables

The maximum length of all conference link cables should not exceed 175 ft and no single run of should exceed 100 ft.

A summary of the cables is shown in the following table. The pin outs for the RJ45 terminated cables 2457-24008-001 and 2457-24009-001 are shown in Specifications. Both of these cables have the same pin out and differ only in length.

Conference Link2 Cables and Descriptions

Clink2 Cable	*Cable Description*
2457-23716-001	RJ45 to Walta connector converter. Typically included with the HDX 9000 series video systems.
2457-23215-001	Walta to Walta cable, 15 ft length
2457-23216-001	Walta to Walta cable, 25 ft length. Included with the Polycom table microphone arrays.
2457-24008-001	RJ45 to RJ45, 50 ft length, cross-over cable. Part of the Polycom ceiling microphone array package.
2457-24009-001	RJ45 to RJ45, 25 ft length, cross-over cable. Part of the Polycom ceiling microphone array extension package.

Conference Link2 Cables and Descriptions

Clink2 Cable	*Cable Description*
2457-24011-001	RJ45 to RJ45, 10 ft length, straight-through cable. Part of the Polycom ceiling microphone array package.
2457-23574-001	RJ45 to RJ45, 18” length, cross-over cable. Included with the SoundStructure device.

Summary

This chapter has described how the Polycom Video Codec conferencing system can be connected to SoundStructure devices over the Conference Link2 interface including a description of the signals and available processing.

In addition, up to four digital microphone arrays may be used with the SoundStructure devices to simplify any audio or video conferencing design.

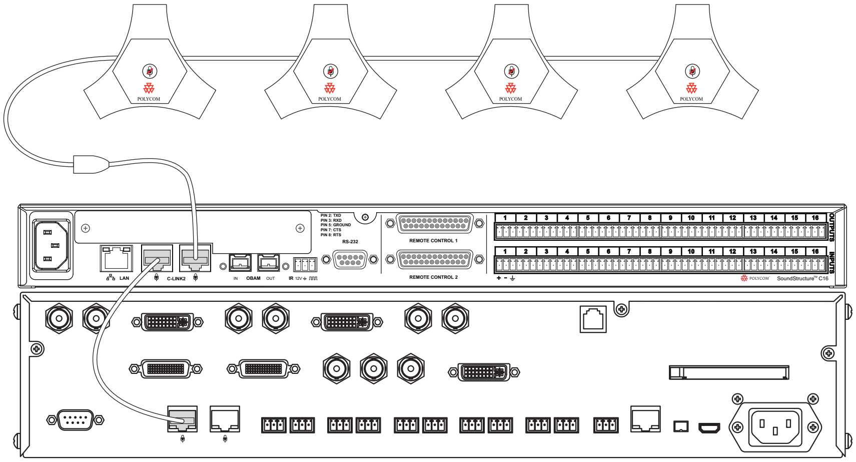
The digital microphone arrays take up the processing of three analog inputs. The following table shows the number of analog inputs that are available based on the number of microphones that are used in a system. As an example, a SoundStructure C16 supports 16 analog inputs. When used with two microphone arrays, 10 analog inputs are still available for use with other analog inputs including microphones, program audio, etc.

Number of Analog Inputs Available for Polycom Video Codecs

		Number of available analog inputs		
		C16	C12	C8
	0	16	12	8
Number of Polycom Microphones	1	13	9	5
	2	10	6	2
	3	7	3	--
	4	4	0	--

The digital microphones should be connected to the right rear CLink2 port and the Polycom Video Codec should be connected to the left CLink2 port as shown in the following figure.

Digital Microphones Connected to a Conference Link2 and Polycom Video Codec



Linking Multiple SoundStructure Devices with One Big Audio Matrix

This chapter describes how to

- link up to eight SoundStructure devices together with One Big Audio Matrix,
- create a configuration file for multiple SoundStructure devices,
- upload and confirm that the SoundStructure system is functional, and
- control a SoundStructure system.

Introduction

Up to eight SoundStructure devices may be linked together using the SoundStructure One Big Audio Matrix (OBAM) interfaces. Each of these SoundStructure devices may also have one telephony card installed for up to 8 telephone cards and support for 16 independent phone lines. Any C-series or SR-series SoundStructure devices may be linked together.

When multiple SoundStructure devices are linked the SoundStructure system displays as one large system with one matrix and one set of input and output channels to configure. SoundStructure Studio version 1.2 or higher and SoundStructure firmware version 1.2 or higher is required to configure a SoundStructure system that is comprised of multiple SoundStructure devices.

Preparing Units for Linking with OBAM

Updating SoundStructure Device Firmware

Before linking multiple SoundStructure devices, the firmware in each SoundStructure device must be updated to version 1.2 or higher. SoundStructure devices with firmware earlier than version 1.2 must be updated one device at a time. For SoundStructure devices with firmware 1.2 and later, the OBAM interface enables multiple SoundStructure devices to have their firmware updated simultaneously.

The steps to update a SoundStructure device's firmware are described in detail in Installing SoundStructure Devices and are summarized here for convenience.

1. Copy the firmware file to a local folder on your computer
2. Connect to the SoundStructure device via Ethernet (recommended) or RS-232 (select a baud-rate of 115,200 due to the size of the firmware file and the subsequent long file transfer times at lower baud rates)
3. Once connected, left-click on the system name to show the firmware update option. Select 'Open' to find the firmware file from your desktop.

4. Select *Update* to begin the firmware update process. Once the firmware has been updated, the SoundStructure system reboots automatically. The front panel light on the SoundStructure device flashes green while booting and turns solid green when the boot process has finished.

Repeat the firmware update process for each SoundStructure device to be linked with OBAM.

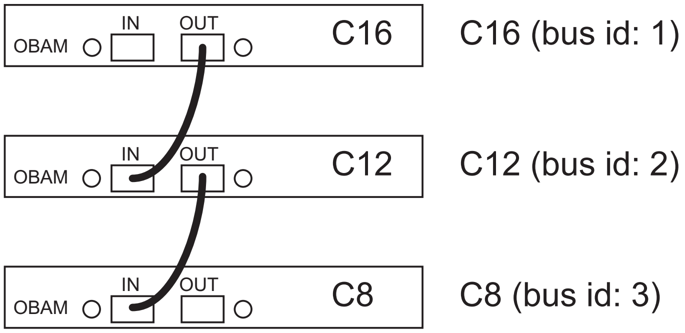
Linking SoundStructure Devices

Once all the SoundStructure devices are running version 1.2 firmware or higher, the next step to link multiple devices together is to Power Down all the SoundStructure devices and then cable OBAM connectors between devices. Connect the *OBAM Out* connector on one device to the *OBAM In* as shown in the following example for a project that utilizes a C16, C12, and C8.

The order the devices are linked is important as it must match the project file that SoundStructure Studio creates. As is described later, if the devices do not match the configuration file when the file is being uploaded, a Convert Project Devices wizard runs to change the devices used in the configuration file to match the actual devices.

A device’s bus ID is automatically assigned to SoundStructure devices based on how the systems are linked. The device that only has a connection on the *OBAM Out* connector has a bus ID of 1 and is referred to as the Master device. The remaining devices are numbered sequentially and are referred to as Slave devices.

Figure: OBAM Linking SoundStructure Devices



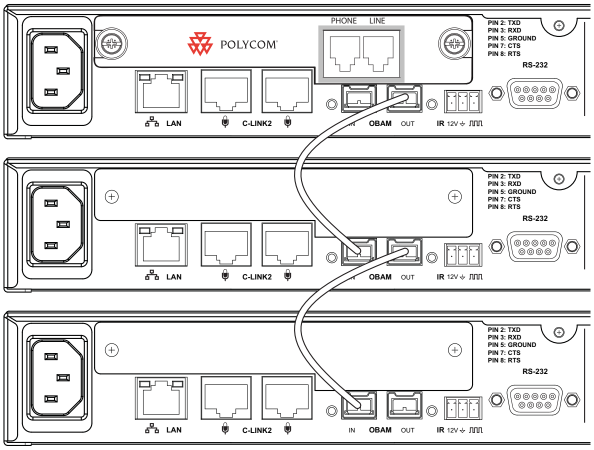
Checking OBAM Cable Length

The OBAM cable supplied is 12 inches long and is designed to connect between SoundStructure devices within an equipment rack. 1394B extenders are not compatible with the OBAM interface.

Installing Multiple Telephony Cards

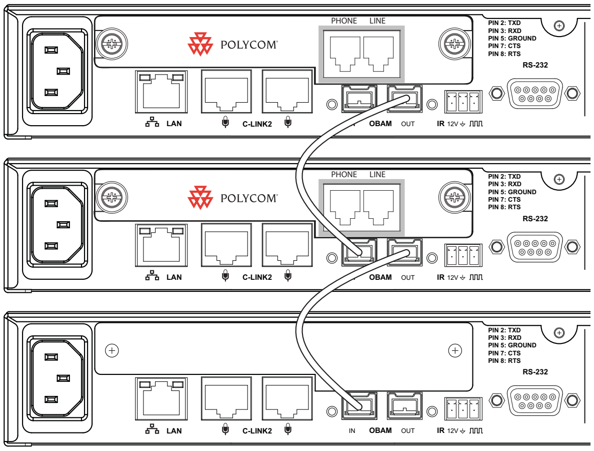
While the devices are powered down, any required telephony cards should be installed into the system. Telephony cards should be installed starting with the Master device and working through the slave devices as shown in the following figure where one TEL1 card is installed into the master device of a three SoundStructure system.

Figure: Installing Telephony Cards


PHONE
LINE
POLYCOM
PIN 2: TXD
PIN 3: RXD
PIN 5: GROUND
PIN 7: CTS
PIN 8: RTS
RS-232
LAN
C-LINK2
IN
OBAM
OUT
IR 12V

If two telephony cards are required, install the second telephony card in the second SoundStructure device as shown in the following figure.

Figure: Installing Multiple Telephony Cards


PHONE
LINE
POLYCOM
PIN 2: TXD
PIN 3: RXD
PIN 5: GROUND
PIN 7: CTS
PIN 8: RTS
RS-232
LAN
C-LINK2
IN
OBAM
OUT
IR 12V

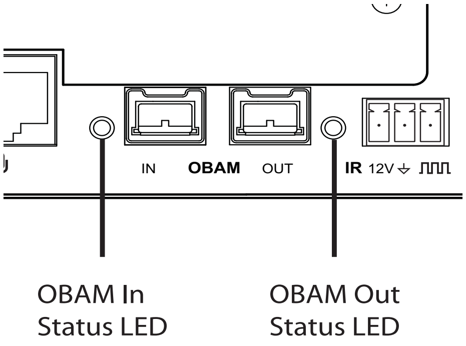
Note: Inserting Telephony Cards

Telephony cards should be inserted into devices starting with the master device and working down the OBAM link (increasing bus IDs) if more than one telephony card is required.

Viewing Rear Panel OBAM LED Status

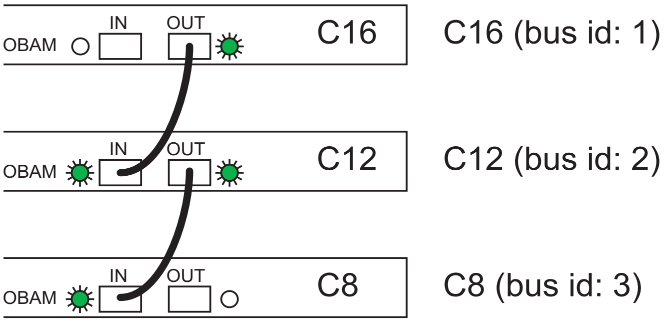
After the devices are linked together, apply power to the SoundStructure devices. After the SoundStructure devices have finished booting, the OBAM status LEDs, shown in the following figure, indicates the status of the OBAM link.

Figure: OBAM Status LED Lights



Under normal operating conditions, the rear panel OBAM Out LED on one SoundStructure device and the OBAM In LED on another SoundStructure device lights up solid green when a valid OBAM Out to OBAM In connection is made as shown in the following figure.

Figure: OBAM Out and OBAM In Status LED Lights



If there is an invalid OBAM connection (OBAM Out to OBAM Out or OBAM In to OBAM In), the OBAM status LEDs blink (0.5 seconds on, 0.5 seconds off).

An OBAM cable may be tested by connecting the cable between the OBAM In and OBAM Out ports on the same SoundStructure system. Depending on the version of firmware, if the cable is functional, the OBAM In and Out LEDs either blink 0.5 seconds on and 0.5 seconds off (version 1.2.0) or the LEDs turn solid green (version 1.2.1 and above). If the LEDs do not light then either the OBAM cable is not functional or there is an error with the OBAM interface on the SoundStructure device.

Viewing Front Panel LED Status

If the linked SoundStructure devices do not have a previously loaded configuration file, the front panel lights on all the SoundStructure devices are solid green after the devices finish booting.

If the OBAM-linked SoundStructure master device has a previously loaded configuration file that does not match the currently linked SoundStructure devices, the front panel LED on all SoundStructure devices turn solid yellow. The solid yellow LED indicates that the SoundStructure project loaded into the master SoundStructure devices does not match the type and number of devices in the overall SoundStructure system. The yellow light turns green once a valid configuration file is loaded into the SoundStructure system as described in the Creating a Multi-Device Configuration File section.

Note: Compatible SoundStructure Configuration Files

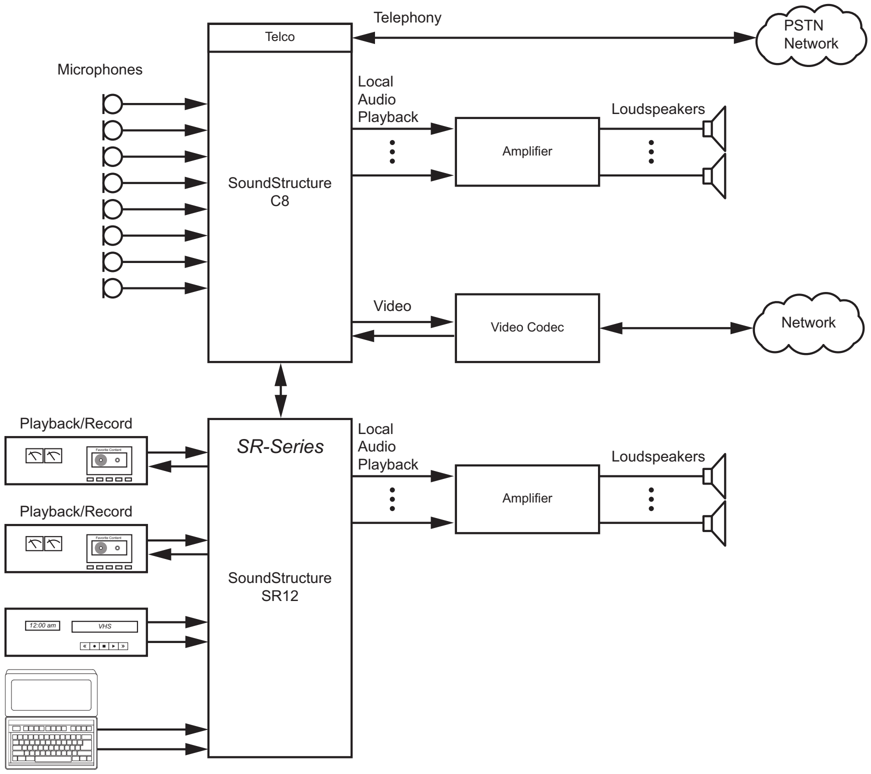
A front panel LED that is solid yellow once devices are linked via OBAM and powered up indicates that the system does not contain a compatible SoundStructure configuration file that matches the number and type of SoundStructure devices linked together. This condition may be corrected by uploading a configuration file that matches the OBAM-linked SoundStructure devices.

Combining C-Series and SR12 Devices

C-series devices may be linked with SR12 devices for up to a total of eight SoundStructure devices. When a SoundStructure SR12 is used with C-series products, the SR12 should be used only to support additional *non*-microphone inputs such as additional program audio sources or to support additional output signals such as for driving additional loudspeaker zones or other outputs as shown in the following figure.

The SoundStructure SR12 does not include acoustic echo cancellation processing on the microphone inputs. Do not use an SR12 to add more conferencing microphones to an installation as the acoustic echoes are not removed by the SR12. Use additional SoundStructure C-series units if necessary to add additional conferencing microphones to a SoundStructure system.

Figure: SR-12 Supporting Additional Output Signals


Telco
Telephony
PSTN Network
Microphones
Local Audio Playback
Loudspeakers
Amplifier
SoundStructure C8
Video
Video Codec
Network
Playback/Record
SR-Series
Local Audio Playback
Loudspeakers
Amplifier
Playback/Record
SoundStructure SR12
12:00 am
VHS

Caution: Do Not Use SR12 to Add Microphones to SoundStructure
Do not use an SR12 to add conferencing microphones to a SoundStructure C-series conferencing design. The SR12 does not include acoustic echo cancellation processing and any microphones used in a conferencing application that are connected to the SR12 has a strong acoustic echo that can only be removed by connecting those microphones to a SoundStructure C-series product.

Creating a Multi-Device Configuration File

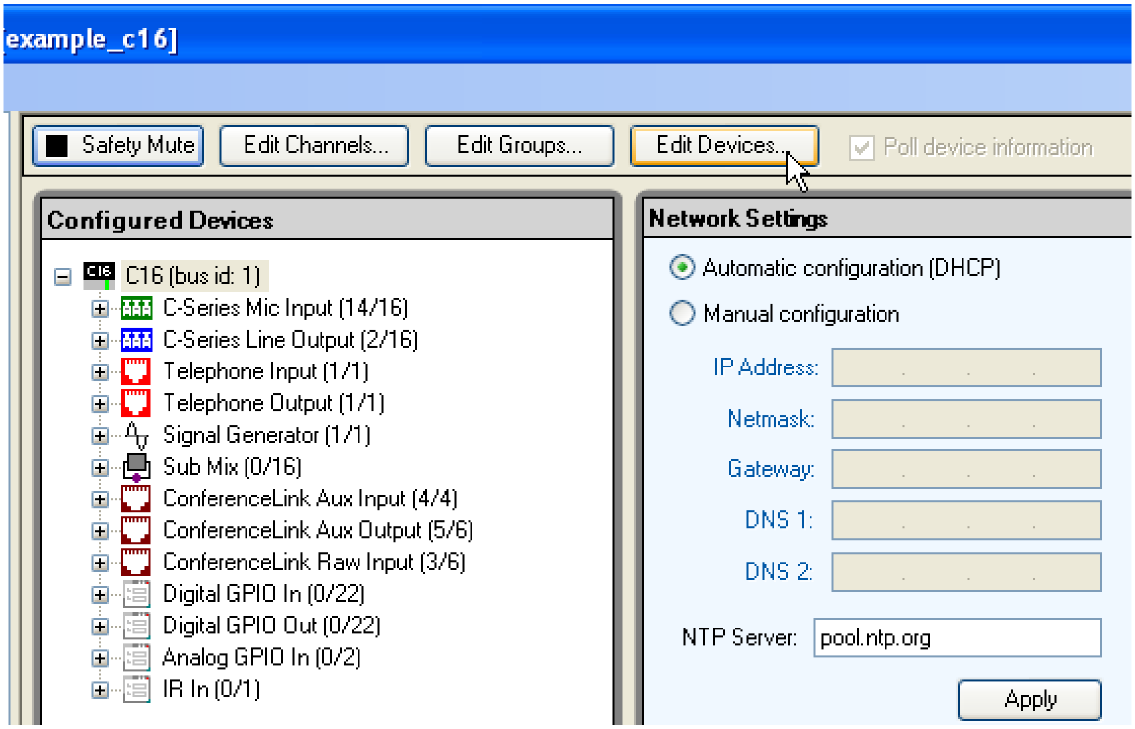
Once the SoundStructure devices are linked and ready to operate as a multi-device system, the next step is to create a configuration file that can be uploaded into the SoundStructure system.

The configuration file for an OBAM-linked system contains information for all the linked SoundStructure devices. SoundStructure Studio is used to upload the configuration into the master device and then the settings are distributed automatically to all the linked units.

Expanding or Contracting an Existing Project

A configuration file for a single device may be expanded to support additional SoundStructure devices or a configuration file for multiple devices may be contracted to operate on fewer SoundStructure devices by opening the source configuration file *offline*, navigating to the Wiring page, and using the Edit Devices button as shown in the following figure. The Edit Devices button is not active when connected *online* to a SoundStructure system.

Figure: Edit Devices on the Wiring Page



C16 (bus id: 1)
- C-Series Mic Input (14/16)
- C-Series Line Output (2/16)
- Telephone Input (1/1)
- Telephone Output (1/1)
- Signal Generator (1/1)
- Sub Mix (0/16)
- ConferenceLink Aux Input (4/4)
- ConferenceLink Aux Output (5/6)
- ConferenceLink Raw Input (3/6)
- Digital GPIO In (0/22)
- Digital GPIO Out (0/22)
- Analog GPIO In (0/2)
- IR In (0/1)

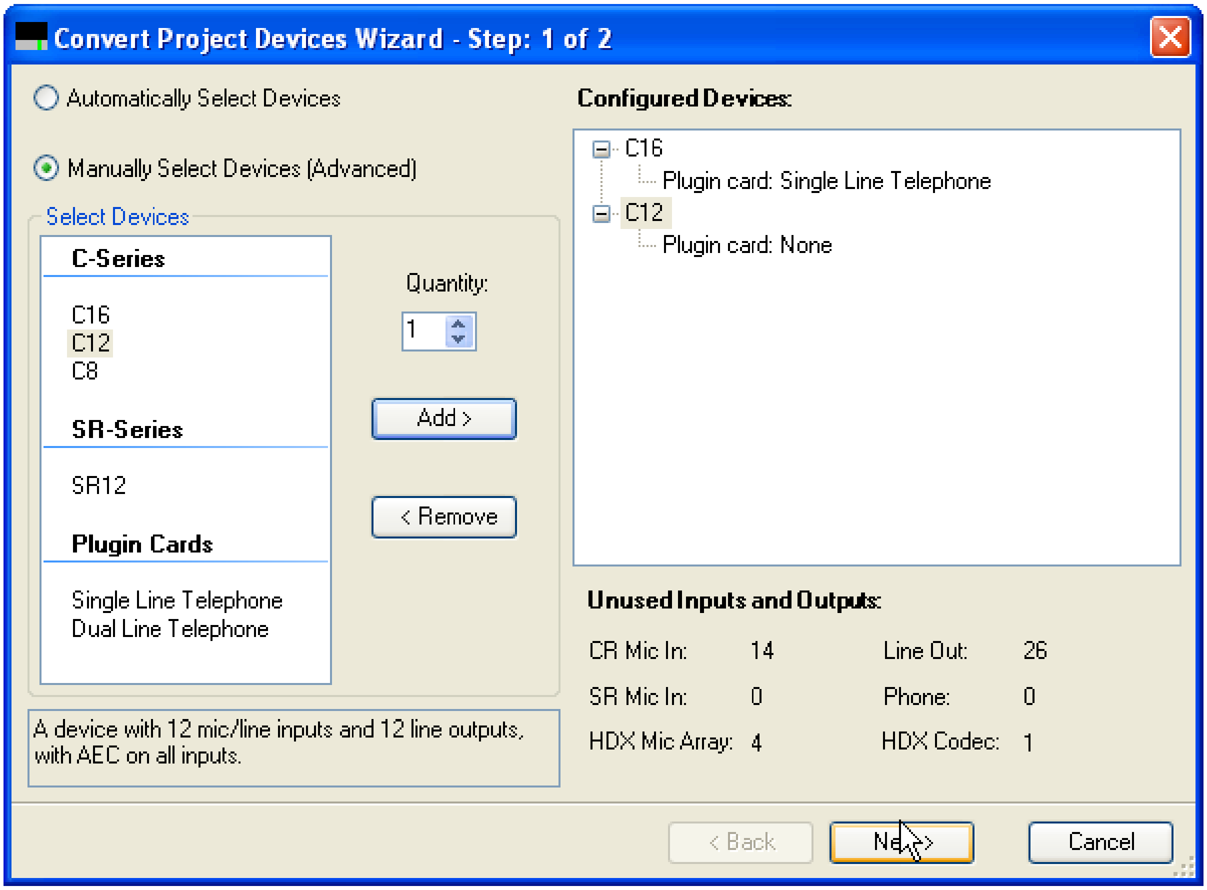
● Automatic configuration (DHCP)
○ Manual configuration

NTP Server: pool.ntp.org

Note: Edit Devices Only Available When Working Offline
The ‘Edit Devices’ button is only active when working offline. Once connected to a SoundStructure system, the ‘Edit Devices’ button is disabled.

The Edit Devices button brings up Convert Projects Device Wizard, a tool for changing devices used in a SoundStructure project. There are two steps to the Convert Projects wizard - a step for selecting the devices and telephony options, and, when downsizing, a step for removing channels. The following figure shows Step 1 in the Convert Projects Devices Wizard where a C12 was added to the system. To add equipment, select the equipment and click **Add**. When finished changing the equipment, press Next to continue to step 2.

Figure: Adding Equipment



○ Automatically Select Devices

● Manually Select Devices (Advanced)

C-Series
C16
C12
C8

SR-Series
SR12

Plugin Cards
Single Line Telephone
Dual Line Telephone

Quantity: 1

A device with 12 mic/line inputs and 12 line outputs, with AEC on all inputs.

Configured Devices:

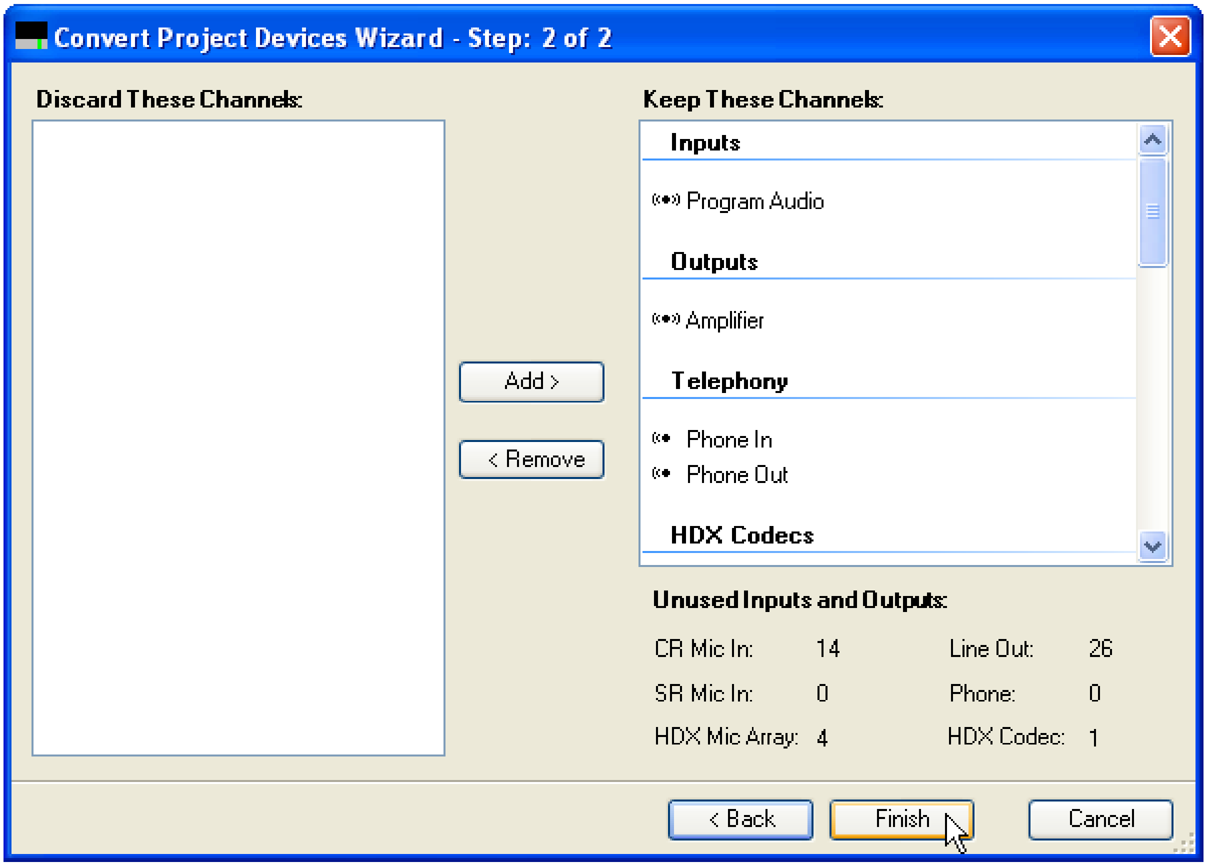
- C16
  - Plugin card: Single Line Telephone
- C12
  - Plugin card: None

Unused Inputs and Outputs:

CR Mic In:	14	Line Out:	26
SR Mic In:	0	Phone:	0
HDX Mic Array:	4	HDX Codec:	1

In the second step of the Convert Project Devices wizard, any channels that no longer fit into the system if the size of the system was reduced must now be removed. If all the channels fit into the new system, as is the case in this example because an entire SoundStructure device was added, the left pane is empty as shown in the following figure.

Figure: New Channels Fitted into a New System



Convert Project Devices Wizard - Step: 2 of 2

Discard These Channels:

Keep These Channels:

Inputs
- Program Audio

Outputs
- Amplifier

Telephony
- Phone In
- Phone Out

HDX Codecs

Unused Inputs and Outputs:

CR Mic In:	14	Line Out:	26
SR Mic In:	0	Phone:	0
HDX Mic Array:	4	HDX Codec:	1

The result of the Edit Devices operation is a *new* configuration file that can be edited. The original device configuration file remains unchanged. Once the device(s) have been added or changed, use the ‘Edit Channels’ button to add more inputs and outputs to the system. Configure the settings for the new channels (AEC reference, equalization, etc.) and then save the settings to a preset and then save the new configuration file to disk.

Make sure the inputs and outputs of the system are physically wired according to how the new devices are configured on Wiring page.

Creating a New Project

A new SoundStructure Studio project may be created for a multi-device system as easily as creating projects for a single SoundStructure device. The steps required to create a file and upload to a collection of SoundStructure devices are listed below.

1. Create a new project using SoundStructure Studio 1.2 or higher by selecting the desired inputs to and outputs from the system in steps 1 and 2 of the design process.

2 On step 3 of the design process, either select the default equipment (if it matches your target devices) or manually change the equipment to match the equipment that you already have. Add the equipment in the order that you have the devices OBAM linked together. In the previous example this would mean adding first a C16, then a C12, and finally a C8.

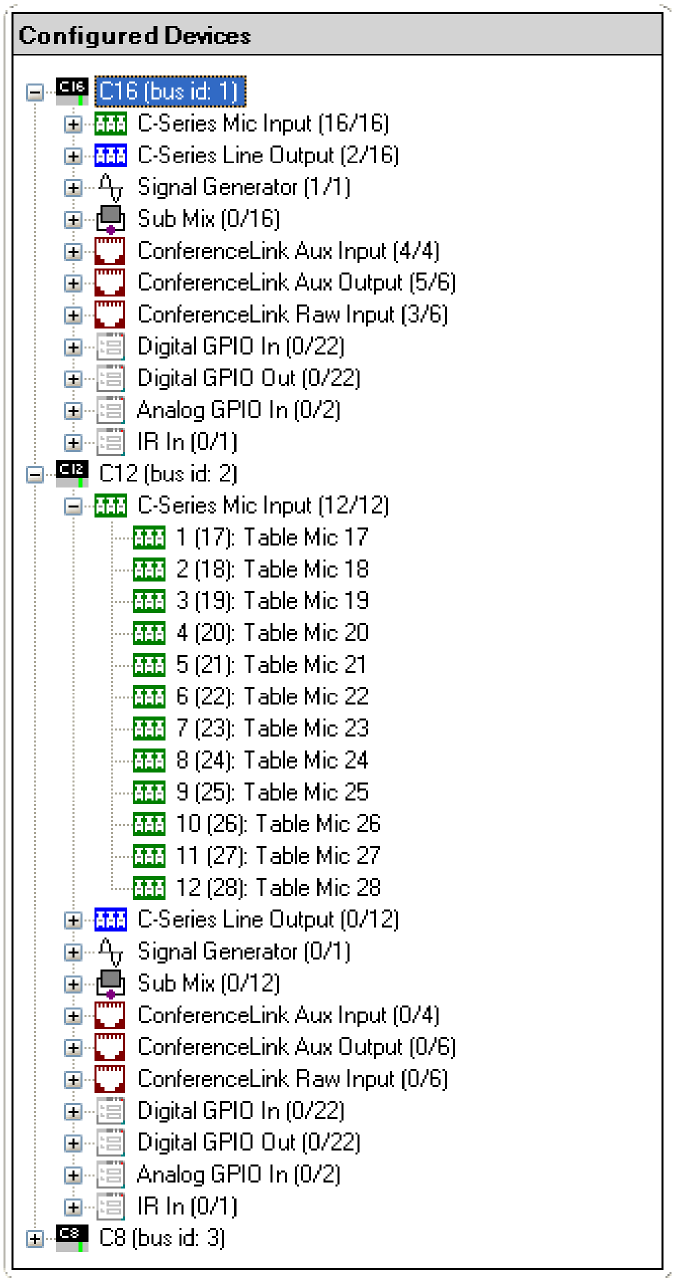
3 On Step 4 of the design process, select Offline configuration.

4 Once the project has been completed, confirm the actual wiring of the system to ensure the physical input and output wiring matches the wiring page.

Viewing Physical Channels on the Wiring Page

As described in Introducing SoundStructure Design Concepts, when multiple SoundStructure devices are linked together, the numbering of the physical channels is sequential across the devices. The wiring page shows the wiring information including both the local input/output numbering and the global input/output numbering as shown in the following figure where the local input/output numbering on the C12 ranges from 1-12 while the global numbering ranges from 17-28 because the C16, as the first device reserves the numbering 1-16.

Figure: Wiring Information for Configured Devices



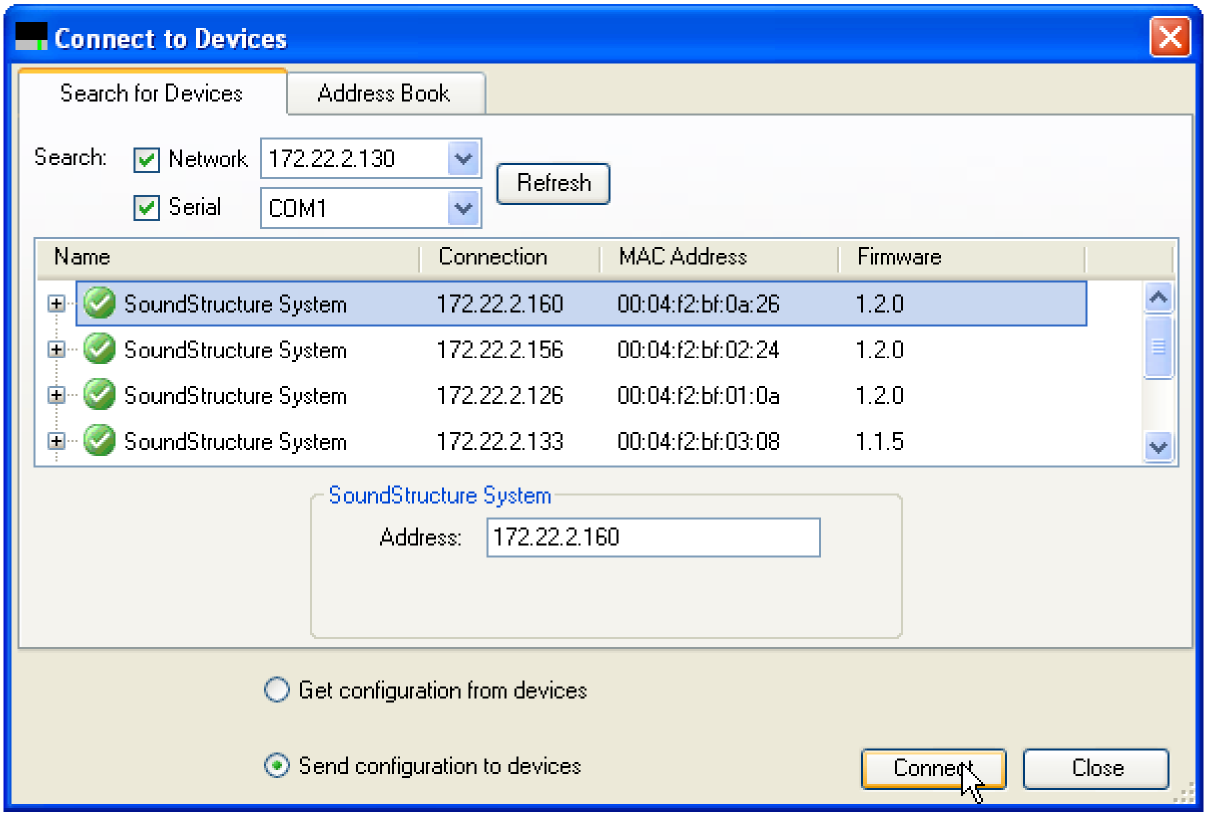
Configured Devices

- C16 (bus id: 1)
  - C-Series Mic Input (16/16)
  - C-Series Line Output (2/16)
  - Signal Generator (1/1)
  - Sub Mix (0/16)
  - ConferenceLink Aux Input (4/4)
  - ConferenceLink Aux Output (5/6)
  - ConferenceLink Raw Input (3/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)
- C12 (bus id: 2)
  - C-Series Mic Input (12/12)
    - 1 (17): Table Mic 17
    - 2 (18): Table Mic 18
    - 3 (19): Table Mic 19
    - 4 (20): Table Mic 20
    - 5 (21): Table Mic 21
    - 6 (22): Table Mic 22
    - 7 (23): Table Mic 23
    - 8 (24): Table Mic 24
    - 9 (25): Table Mic 25
    - 10 (26): Table Mic 26
    - 11 (27): Table Mic 27
    - 12 (28): Table Mic 28
  - C-Series Line Output (0/12)
  - Signal Generator (0/1)
  - Sub Mix (0/12)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)
- C8 (bus id: 3)

Uploading Configuration Files

To upload a configuration to the SoundStructure system, open the configuration file within SoundStructure Studio. Select **Connect to Devices** and find the SoundStructure system to receive the configuration file as shown in the following figure.

Figure: Connecting to SoundStructure Devices



Search: ☑ Network 172.22.2.130

☑ Serial COM1

Name	Connection	MAC Address	Firmware
SoundStructure System	172.22.2.160	00:04:f2:bf:0a:26	1.2.0
SoundStructure System	172.22.2.156	00:04:f2:bf:02:24	1.2.0
SoundStructure System	172.22.2.126	00:04:f2:bf:01:0a	1.2.0
SoundStructure System	172.22.2.133	00:04:f2:bf:03:08	1.1.5

SoundStructure System

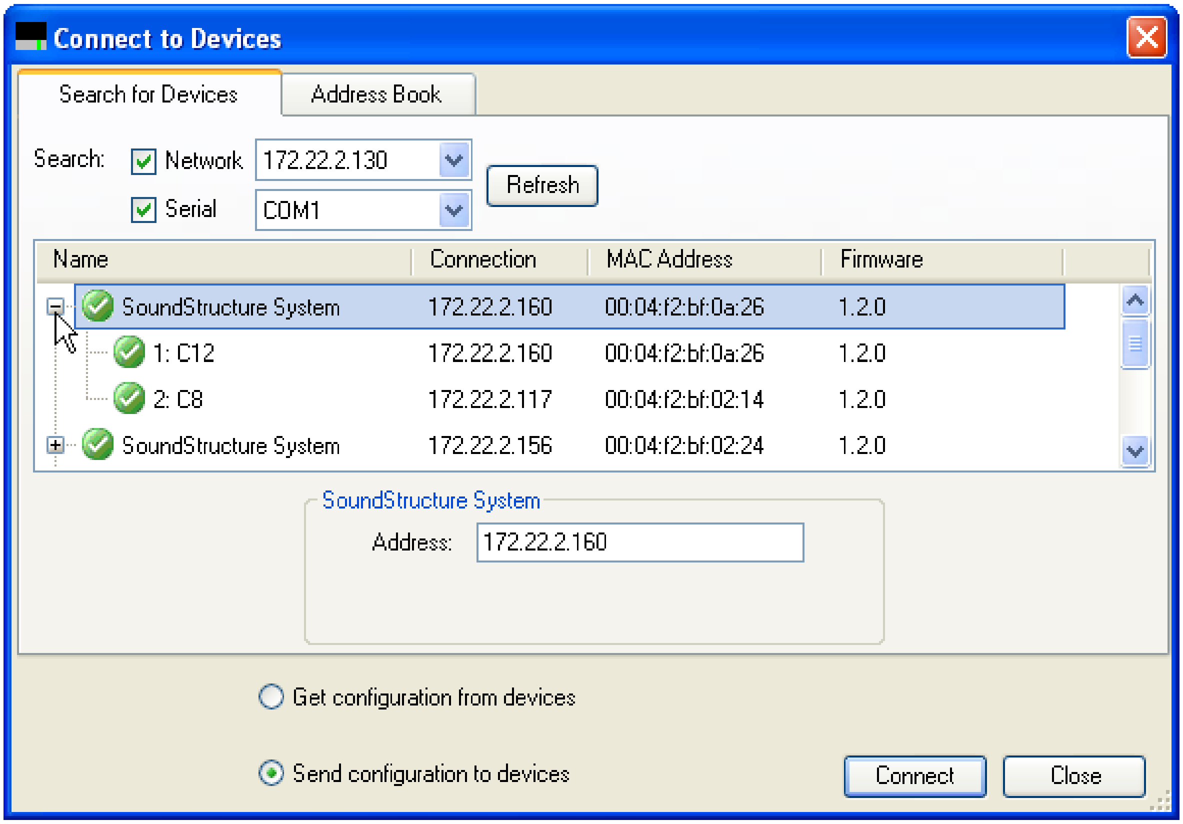
Address: 172.22.2.160

○ Get configuration from devices

● Send configuration to devices

Any detected SoundStructure systems may be expanded to show the individual devices that are part of the system by clicking the “+” sign next to the system name. The result displays as shown in the following figure.

Figure: **Viewing Individual Devices of a SoundStructure System**


Connect to Devices
Search for Devices
Address Book
Refresh
SoundStructure System
Connect
Close

Search: ☑ Network 172.22.2.130

☑ Serial COM1

Name	Connection	MAC Address	Firmware
SoundStructure System	172.22.2.160	00:04:f2:bf:0a:26	1.2.0
1: C12	172.22.2.160	00:04:f2:bf:0a:26	1.2.0
2: C8	172.22.2.117	00:04:f2:bf:02:14	1.2.0
SoundStructure System	172.22.2.156	00:04:f2:bf:02:24	1.2.0

Address: 172.22.2.160

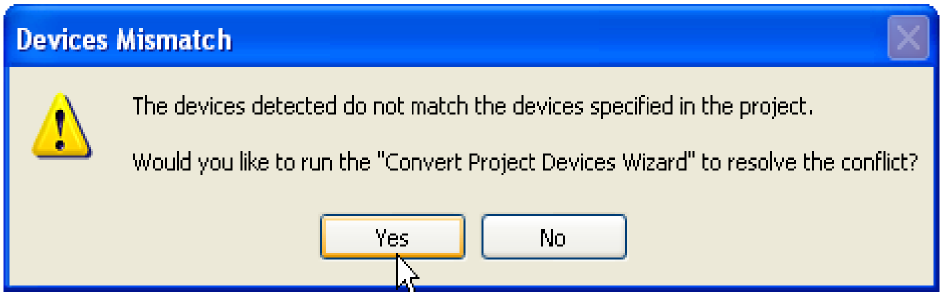
○ Get configuration from devices

● Send configuration to devices

To send the configuration file to the SoundStructure system, select **Send configuration to devices** and press **Connect**.

If the actual SoundStructure target devices do not match the devices in the configuration file, SoundStructure Studio presents the option of either correcting the mismatch or not uploading the configuration file as shown in the following figure.

Figure: Devices Mismatch Dialog



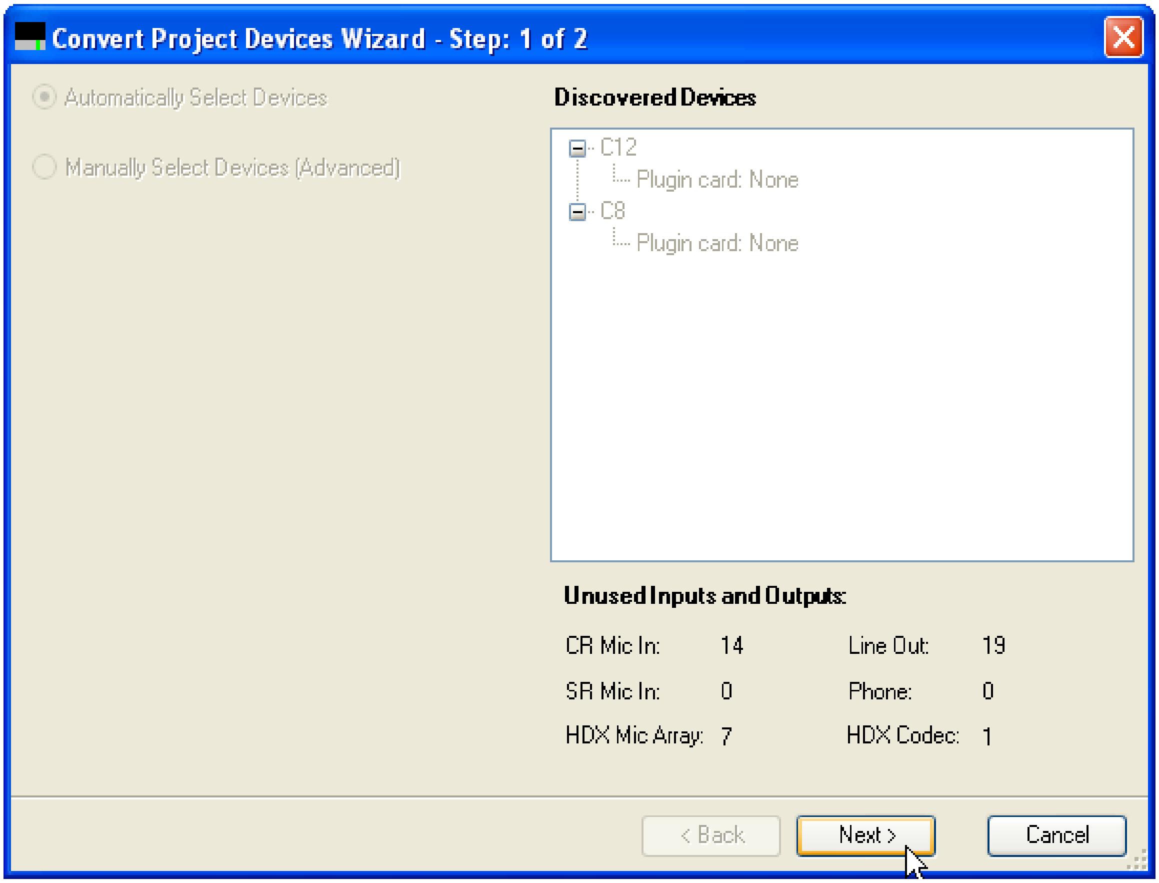
Devices Mismatch

The devices detected do not match the devices specified in the project.

Would you like to run the "Convert Project Devices Wizard" to resolve the conflict?

Select **No** to cancel the project upload or select **Yes** to correct the device mismatch. Selecting **Yes** shows the following dialog that shows the discovered devices and the unused inputs and outputs if this equipment is used for the configuration file.

Figure: Discovered Devices and Unused Inputs and Outputs



Convert Project Devices Wizard - Step: 1 of 2

● Automatically Select Devices

○ Manually Select Devices (Advanced)

Discovered Devices

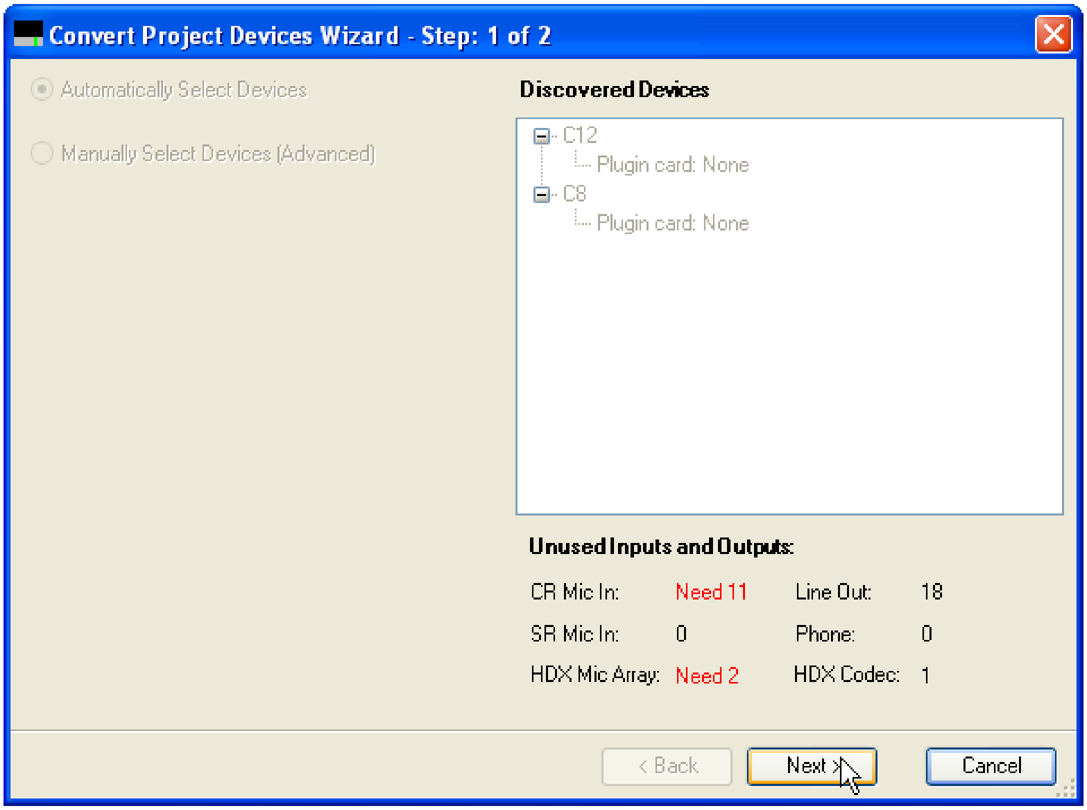
- C12
  - Plugin card: None
- C8
  - Plugin card: None

Unused Inputs and Outputs:

CR Mic In:	14	Line Out:	19
SR Mic In:	0	Phone:	0
HDX Mic Array:	7	HDX Codec:	1

If the target equipment does not support all the inputs and outputs that the project requires as shown in the following figure, then the project must either be scaled back or the number or type of target devices increased.

Figure: Required Inputs and Outputs for Target Equipment



Convert Project Devices Wizard - Step: 1 of 2

○ Automatically Select Devices

○ Manually Select Devices (Advanced)

Discovered Devices

- C12
  - Plugin card: None
- C8
  - Plugin card: None

Unused Inputs and Outputs:

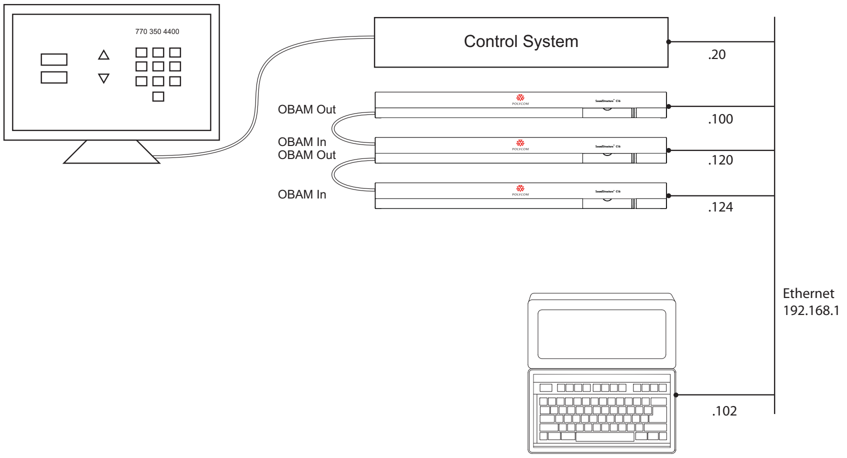
CR Mic In:	Need 11	Line Out:	18
SR Mic In:	0	Phone:	0
HDX Mic Array:	Need 2	HDX Codec:	1

To add more devices, follow the steps outlined earlier in this chapter in the Expanding or Contracting an Existing Project section. If the size of the system is reduced, channels may need to be removed as described previously.

Controlling the SoundStructure System

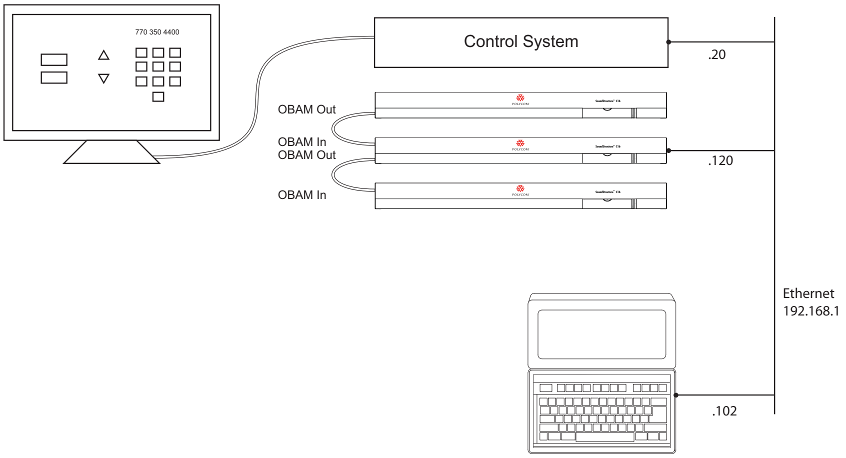
Only one control port (RS-232 or Ethernet) is required to control a collection of SoundStructure systems. If multiple SoundStructure devices within a SoundStructure system are connected to the local network, as shown in the following figure the IP address of the system is the address of the master or, if the master doesn’t have an IP connection, the system closest to the master that does have an IP connection. In the following example, the IP address of the system is 192.168.1.100 as that is the IP address of the master SoundStructure device.

Figure: Master Device IP Address Example


770 350 4400
Control System
.20
OBAM Out
.100
OBAM In
OBAM Out
.120
OBAM In
.124
Ethernet
192.168.1
.102

In the following figure, the address of the system is 192.168.1.120.

Figure: Example Address System 192.168.1.120


770 350 4400
Control System
.20
OBAM Out
OBAM In
OBAM Out
.120
OBAM In
Ethernet
192.168.1
.102

If there are multiple IP addresses associated the different devices in a SoundStructure system, it is possible to connect to the system via any of the IP addresses although SoundStructure Studio presents the overall system IP address when the system is discovered.

Accessing SoundStructure Logs

When accessing the logs of a SoundStructure system, the master device logs contain all the command and acknowledgment information for the entire SoundStructure system. When logs are requested, the logs are retrieved from the master SoundStructure device.

Connecting Polycom Microphones

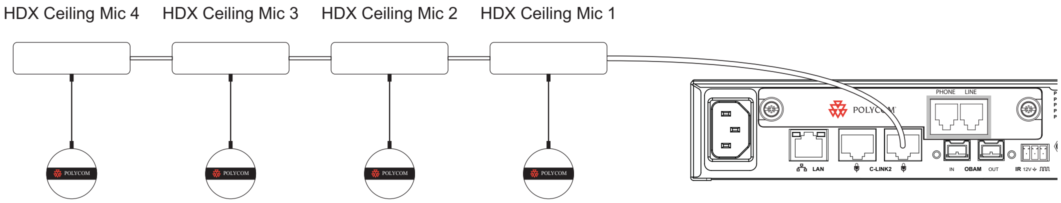
As described in Connecting Over Conference Link2, up to four Polycom microphones may be connected to each SoundStructure device depending on the SoundStructure model. With OBAM linked devices, a total of 32 Polycom microphones may be added to a system of eight SoundStructure devices.

Note: Updating Polycom Microphones’ Firmware

When using Polycom microphones, update the Polycom microphones to the latest firmware by connecting each microphone one at a time to the SoundStructure device. The SoundStructure device compares the version of microphone firmware in the SoundStructure device with the firmware in the microphone. If the SoundStructure device contains a newer version of microphone firmware, the Polycom microphone is automatically updated with the new firmware.

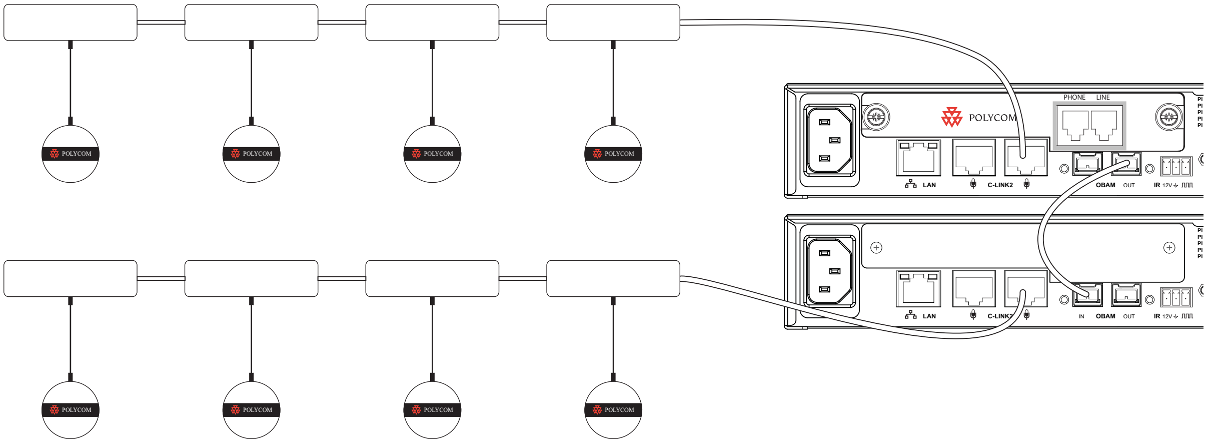
Microphones are numbered sequentially across SoundStructure devices as shown in the following figure with microphones plugged into the right rear CLink2 port. As discussed in Connecting Over Conference Link2, Polycom digital microphones plugged into the SoundStructure’s right rear Clink2 port are numbered so that the closest microphone corresponds to the first Polycom microphone from SoundStructure Studio’s perspective.

Figure: Polycom Microphone Numbering



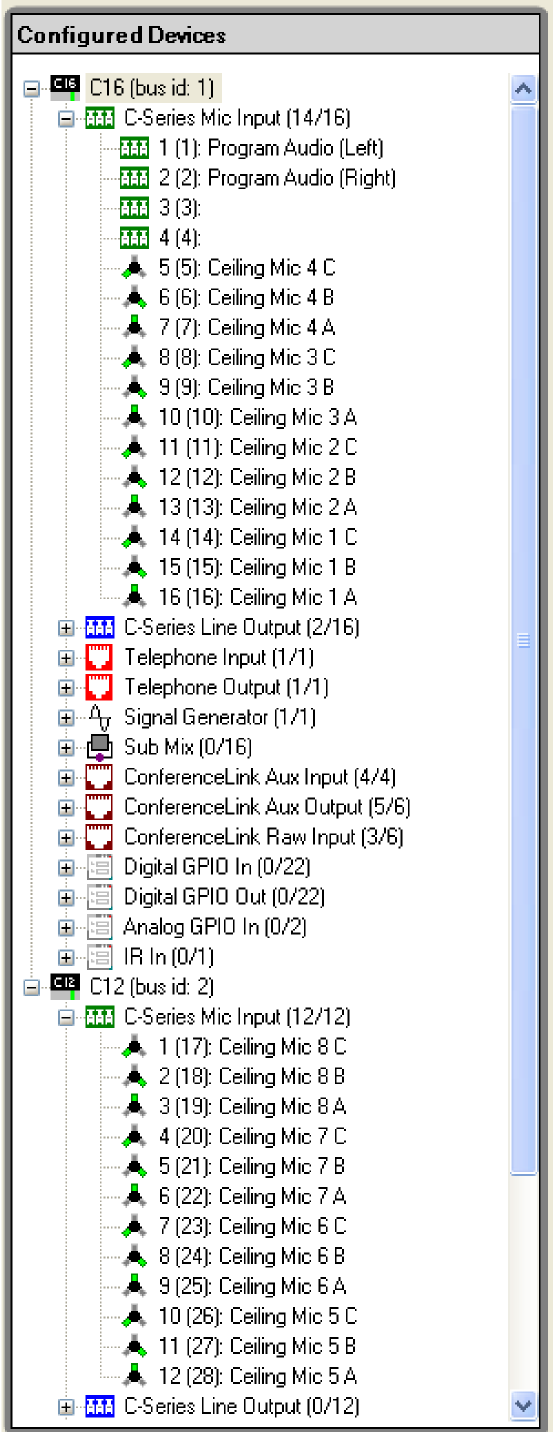
When Polycom microphones are connected across multiple SoundStructure devices, the same numbering sequence applies. For example, designing a large system with eight Polycom microphones may be wired as shown in the next figure with the first four microphones connected to the master SoundStructure device and the next four microphones connected to the slave SoundStructure device.

Figure: Connecting Multiple Polycom Microphones


POLYCOM
POLYCOM
POLYCOM
POLYCOM
POLYCOM
PHONE
LINE
LAN
C-LINK2
OBAM
OUT
IR
12V
POLYCOM
POLYCOM
POLYCOM
POLYCOM
LAN
C-LINK2
IN
OBAM
OUT
IR
12V

The Wiring page for this configuration is shown in the following figure with the first four Polycom microphones on the SoundStructure C16 (bus id 1) and the remaining four Polycom microphones on SoundStructure C12 (bus id 2).

Figure: Wiring Information for Microphone Configuration



Configured Devices

- C16 (bus id: 1)
  - C-Series Mic Input (14/16)
    - 1 (1): Program Audio (Left)
    - 2 (2): Program Audio (Right)
    - 3 (3):
    - 4 (4):
    - 5 (5): Ceiling Mic 4 C
    - 6 (6): Ceiling Mic 4 B
    - 7 (7): Ceiling Mic 4 A
    - 8 (8): Ceiling Mic 3 C
    - 9 (9): Ceiling Mic 3 B
    - 10 (10): Ceiling Mic 3 A
    - 11 (11): Ceiling Mic 2 C
    - 12 (12): Ceiling Mic 2 B
    - 13 (13): Ceiling Mic 2 A
    - 14 (14): Ceiling Mic 1 C
    - 15 (15): Ceiling Mic 1 B
    - 16 (16): Ceiling Mic 1 A
  - C-Series Line Output (2/16)
  - Telephone Input (1/1)
  - Telephone Output (1/1)
  - Signal Generator (1/1)
  - Sub Mix (0/16)
  - ConferenceLink Aux Input (4/4)
  - ConferenceLink Aux Output (5/6)
  - ConferenceLink Raw Input (3/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)
- C12 (bus id: 2)
  - C-Series Mic Input (12/12)
    - 1 (17): Ceiling Mic 8 C
    - 2 (18): Ceiling Mic 8 B
    - 3 (19): Ceiling Mic 8 A
    - 4 (20): Ceiling Mic 7 C
    - 5 (21): Ceiling Mic 7 B
    - 6 (22): Ceiling Mic 7 A
    - 7 (23): Ceiling Mic 6 C
    - 8 (24): Ceiling Mic 6 B
    - 9 (25): Ceiling Mic 6 A
    - 10 (26): Ceiling Mic 5 C
    - 11 (27): Ceiling Mic 5 B
    - 12 (28): Ceiling Mic 5 A
  - C-Series Line Output (0/12)

A Polycom digital microphone may be moved between SoundStructure devices by moving the individual elements on the wiring page and also by connecting the Polycom digital microphone physically to a different SoundStructure device. The three elements A, B, and C, of each Polycom digital microphones must all reside on the same SoundStructure device where the digital microphone is plugged into. It is not possible to allocate a digital microphone’s elements across SoundStructure device boundaries where for example element A is one SoundStructure device and element B and C are on a different SoundStructure system.

As with any application that uses Polycom microphones, update the microphones to the latest microphone firmware by connecting each microphone, one at a time, to the SoundStructure device. If there is an Polycom Video Codec in the system, disconnect the Polycom Video Codec from the SoundStructure device’s other CLink2 port. The SoundStructure device compares the version of firmware in the SoundStructure device with that in the microphone and if the SoundStructure device contains a newer version of microphone firmware, the Polycom microphone is updated with the new firmware. This process takes less than 10 seconds per microphone and while the microphone firmware is being updated, the LED’s inside the Polycom microphone turns orange.

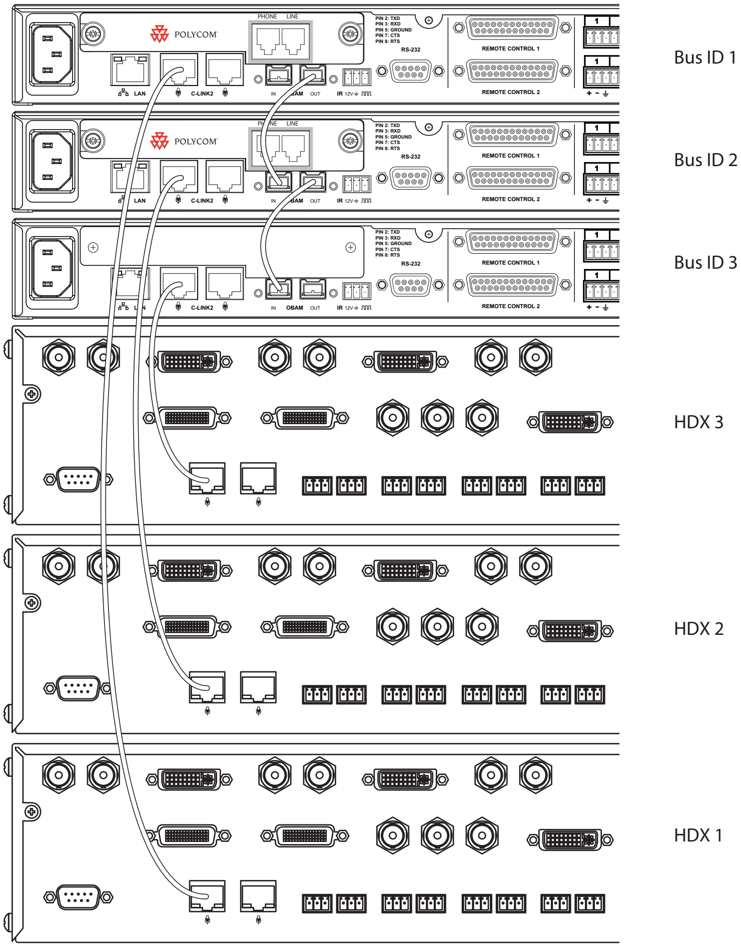
Connecting Multiple Polycom Video Codec Conferencing Systems

Using OBAM -linked SoundStructure devices, it is possible for each SoundStructure device to be digitally connected to a Polycom Video Codec conferencing system as shown in the following figure.

Each Polycom Video Codec requires one connection to an available CLink2 port on the rear panel of a SoundStructure device and only one Polycom Video Codec may be connected per SoundStructure device. The first Polycom Video Codec should be connected to the master SoundStructure device and then subsequent Polycom Video Codecs connected to other SoundStructure devices. In the following figure, Codec 1 is connected to the master SoundStructure (Bus ID 1), Codec 2 is connected to the SoundStructure with Bus ID 2, and Codec 3 is connected to the SoundStructure with bus ID 3.

Specifications includes details for the pin outs of the Clink2 cable required to connect between the Polycom Video Codec and SoundStructure devices.

Figure: Digitally Connecting Polycom Video Codecs



As described in Connecting Over Conference Link2, when a Polycom Video Codec is muted, the codec sends a command to mute the virtual channel or group with the name "Mics". When using multiple Polycom Video Codecs over Clink2, if any Polycom Video Codec is told to mute via a button press on a microphone, an IR key press, or a control system command to the Polycom Video Codec, then the channels defined by "Mics" are muted within SoundStructure. Volume up and volume down operate in a similar manner when any Polycom Video Codec receives a volume up or volume down command, the SoundStructure receives a command to adjust the fader of the "Amplifier" virtual channel.

If multiple Polycom Video Codecs are being used independently within a SoundStructure system, ensure that the SoundStructure system does not include virtual channel names "Mics" or "Amplifier" or if those names are used, ensure that they are defined in such a way that the system operates as desired. An easy way to customize the definition of "Mics" and "Amplifier" virtual channels is to define submixes with the name

“Mics” and “Amplifier” and then use presets or partial presets to mute and unmute the desired signals to these submixes to achieve the desired behavior when a particular Polycom Video Codec is muted or has volume adjusted. If the virtual channel names “Mics” and “Amplifier” are not defined, then nothing is muted and no volume is adjusted on the SoundStructure when one of the Polycom Video Codecs has its mute status changed or volume adjusted.

Installing SoundStructure Devices

This chapter describes how to take the SoundStructure designs created in Chapters 4 and 5 and upload and confirm that the system is fully functional. Once the SoundStructure design has been created, the next steps are to match the physical wiring of the system, upload the settings, make final adjustments to the system, and save the settings to a preset.

For information on rack mounting SoundStructure devices or terminating any of the connectors such as the analog input and output signals refer to the SoundStructure Hardware Installation Guide or Creating Advanced Applications in this manual.

Configuration Files

Configuration files store all the settings associated with a SoundStructure project including the system name, the devices and plug-in cards used in the design, the virtual channel definitions, default channel settings, Ethernet and RS-232 settings, current device settings, and presets. Configuration files have an STR extension and are stored as binary files. The basic configuration file structure is shown in the following figure.

Basic SoundStructure Configuration File Structure

SoundStructure Configuration File

SoundStructure Devices
Virtual Channel and Group Definitions
Default Parameter Values
Ethernet and RS-232 Settings
Current Settings
Presets

Configuration files are both saved to disk when a File Save option is executed from SoundStructure Studio.

Any changes to the device settings that need to survive a power cycle should be saved to presets with the Preset Save operation as described later in this chapter.

Wiring The Devices

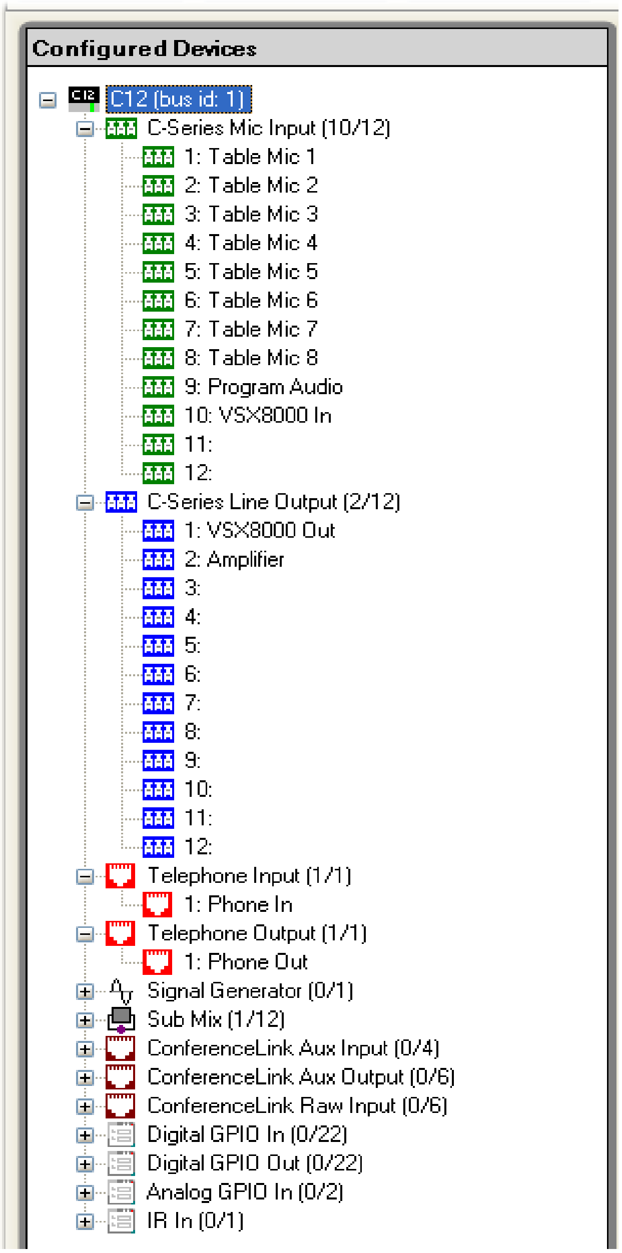
One of the most important steps when working with SoundStructure devices is to ensure the physical cabling (for instance what’s plugged into input 3) of the system *exactly* matches how the virtual channels are defined.

Virtual channels, as introduced in Introducing SoundStructure Design Concepts, provide an abstraction layer around the physical input and output channels. Virtual channels make it possible to refer to, and control, signals by their virtual channel names rather than by the physical input and output numbers. Virtual channels make the system more portable as control system code that is developed can be reused by using the same virtual channel names across different installations - regardless of how the system is physically cabled.

As a system is being designed with SoundStructure Studio, the SoundStructure Studio software defines the virtual channels and then uses the virtual channels with all subsequent operations on those channels.

The first step in verifying the wiring is to view the wiring page within SoundStructure Studio and expand the inputs and outputs as shown in the following figure.

Device Wiring Information



Configured Devices

- C12 (bus id: 1)
  - C-Series Mic Input (10/12)
    - 1: Table Mic 1
    - 2: Table Mic 2
    - 3: Table Mic 3
    - 4: Table Mic 4
    - 5: Table Mic 5
    - 6: Table Mic 6
    - 7: Table Mic 7
    - 8: Table Mic 8
    - 9: Program Audio
    - 10: VSX8000 In
    - 11:
    - 12:
  - C-Series Line Output (2/12)
    - 1: VSX8000 Out
    - 2: Amplifier
    - 3:
    - 4:
    - 5:
    - 6:
    - 7:
    - 8:
    - 9:
    - 10:
    - 11:
    - 12:
  - Telephone Input (1/1)
    - 1: Phone In
  - Telephone Output (1/1)
    - 1: Phone Out
  - Signal Generator (0/1)
  - Sub Mix (1/12)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

The wiring page shows the definitions of the virtual channels along with the underlying physical channels. In this figure table microphones 1 through 8 are connected to physical inputs 1 through 8, the program audio is connected to input 9 and the VSX8000 input is connected to input 10. On the outputs, the amplifier output is connected to physical output 2 and the VSX8000 output channel is connected to physical output 1.

If the system were wired incorrectly and the VSX8000 Out channel and Amplifier channel were reversed due to a physical wiring error, then the signals that were routed to the VSX8000 output channel would now be physically connected to the amplifier. This type of problem could cause the system to immediately generate feedback into the room since the microphones would be routed unintentionally to the amplifier rather than to the codec - a result that is certainly not desired! This example underscores the importance of ensuring the physical connections are the same as the SoundStructure devices expect.

The CAD drawing that corresponds to this wiring page is shown in the following figure.

CAD Drawing for Wiring Page Information

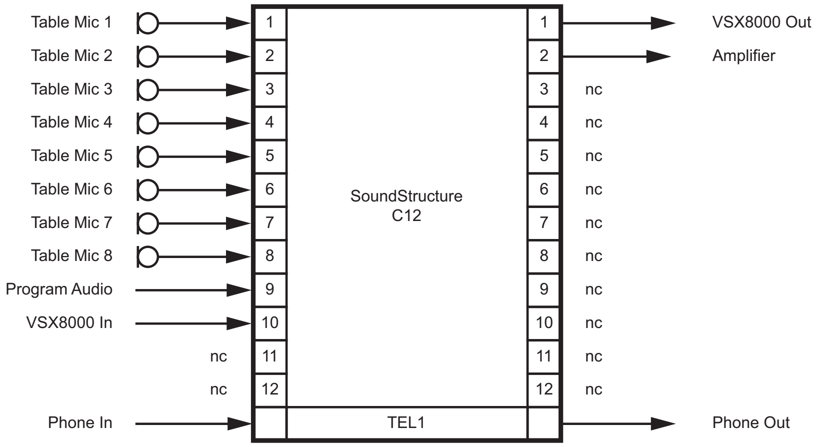


Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
Program Audio
VSX8000 In
nc
nc
Phone In
1
2
3
4
5
6
7
8
9
10
11
12
SoundStructure
C12
TEL1
VSX8000 Out
Amplifier
nc
Phone Out

Note: Physical Wiring Must Match Virtual Wiring

The physical wiring of a system must match the virtual wiring page definition or the system does not operate properly.

There are two options if the actual system wiring doesn’t match the wiring defined by SoundStructure Studio:

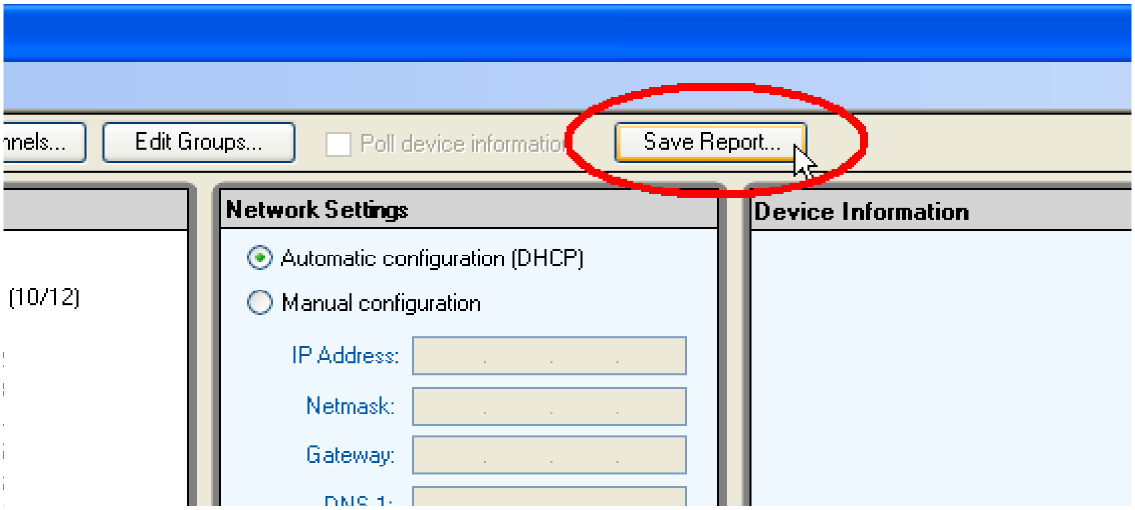
1. Rewire the system physically
2. Rewire the system virtually

Rewiring the system physically requires access to the equipment rack, ensuring the physical cables can still reach their new locations, and moving rear-panel phoenix connectors.

Rewiring the system virtually requires moving signals on the wiring page. This can be done by clicking and dragging the virtual channels signals to their desired inputs and outputs. It is generally simpler to move the virtual signals than the physical wiring.

There is a wiring report that can be created by clicking **Save Report** on the wiring page as shown in the following figure.

Saving Wiring Reports


nnels...
Edit Groups...
☐ Poll device informatio
Save Report...
Network Settings
Device Information
(10/12)
● Automatic configuration (DHCP)
○ Manual configuration
IP Address:
Netmask:
Gateway:
DNS 1:

The wiring report for this system is shown next.

```
SoundStructure system: SoundStructure System
C12 (bus id: 1)
C-Series Mic Input
1: Table Mic 1
2: Table Mic 2
3: Table Mic 3
4: Table Mic 4
5: Table Mic 5
6: Table Mic 6
7: Table Mic 7
8: Table Mic 8
9: Program Audio
10: VSX8000 In

C-Series Line Output
1: VSX8000 Out
2: Amplifier

Plugin Card: Single Line Telephone
1: Phone In, Phone Out
```

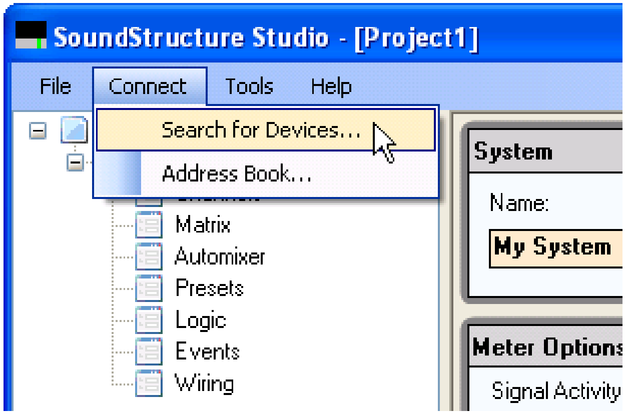
Once the signal wiring is completed, the next step is to upload the settings to the device.

Uploading A Configuration File

Configuration files are uploaded to a SoundStructure device or downloaded from a SoundStructure device by using the SoundStructure Studio software.

To upload a configuration file to the SoundStructure devices, first open the SoundStructure Studio design file and then select **Connect** > **Search for Devices** as shown in the following figure.

Searcing for Devices


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Search for Devices...
Address Book...
Matrix
Automixer
Presets
Logic
Events
Wiring
System
Name:
My System
Meter Options
Signal Activity

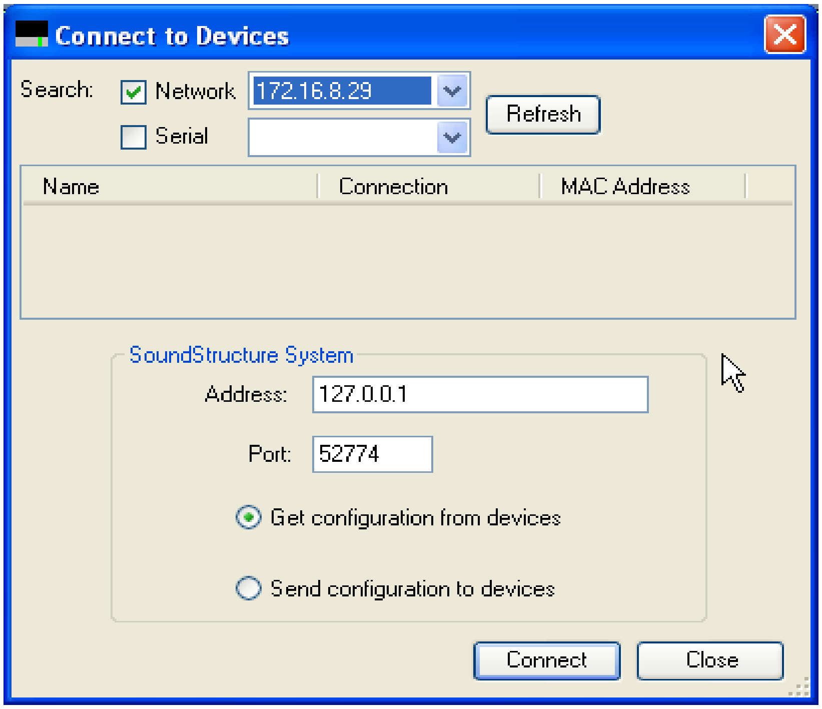
This selection makes the **Connect to Devices** window display as shown in the next figure. There are two ways to connect to the SoundStructure device: through the RS-232 and through the network interface. Select the check box next to the interface to use for the upload or download.

When the check box next to the Network interface is selected, SoundStructure Studio queries which devices are on the network. All devices on the same subnet as the Ethernet interface are displayed by default.

The SoundStructure system names that are found are shown with their System Name (see Managing SoundStructure Systems for information on how to set the system name), IP address or serial port, and MAC address. The MAC address may be found by looking inside the front panel door on the SoundStructure device.

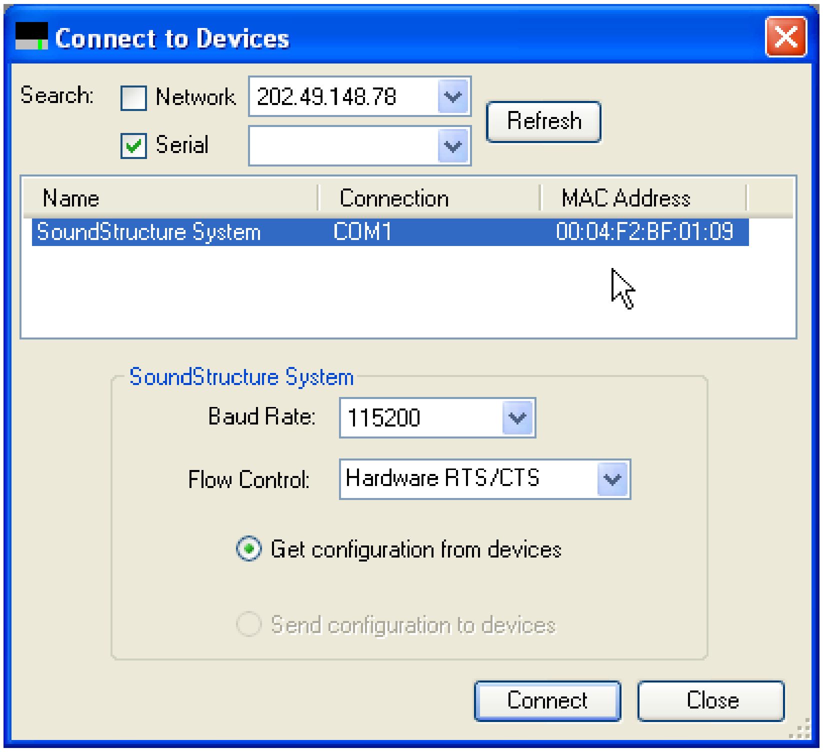
Select the device to upload the file to and select **Send configuration to devices** and **Connect**. The Send configuration to devices option is only enabled if there is a valid configuration file open in SoundStructure Studio.

Connecting to a SoundStructure Device


Connect to Devices
Search:
☑ Network
172.16.8.29
☐ Serial
Refresh
Name
Connection
MAC Address
SoundStructure System
Address: 127.0.0.1
Port: 52774
● Get configuration from devices
○ Send configuration to devices
Connect
Close

If the Serial control is checked, the system also searches for devices over the RS-232 interface as shown in the following figure. Any discovered devices are displayed and the baud-rate and flow control settings required to connect to those devices are displayed.

Searching for Devices over the RS-232 Interface



Search: ☐ Network 202.49.148.78

☑ Serial

Name	Connection	MAC Address
SoundStructure System	COM1	00:04:F2:BF:01:09

SoundStructure System

Baud Rate: 115200

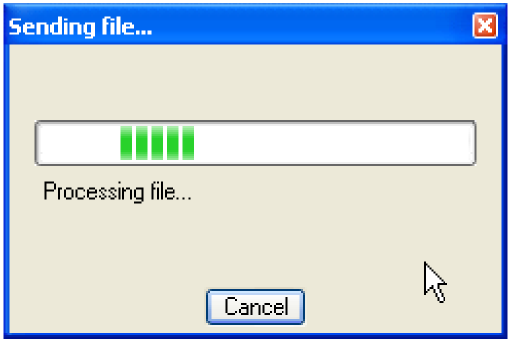
Flow Control: Hardware RTS/CTS

● Get configuration from devices

○ Send configuration to devices

Once the device is selected, a transfer window opens as shown in the following figure showing the state of the file transfer.

Sending File Process Dialog



Processing file...

If the device is running a configuration file that had previously been uploaded, the output channels are muted while the new configuration is uploaded. The audio is unmuted after the upload of the configuration file has been completed.

Once the file has been uploaded, the settings are stored in the non-volatile memory of the device.

Downloading A Configuration File

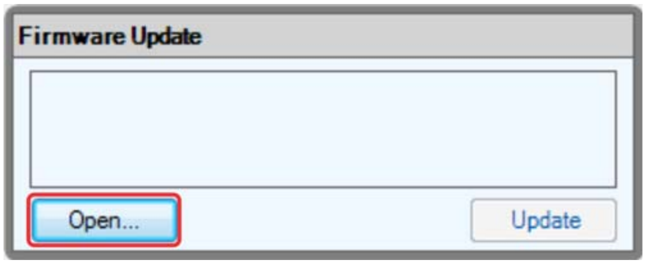
As with uploading a configuration file, downloading a configuration file from a SoundStructure device to SoundStructure Studio involves selecting the Connect to Devices menu option, selecting the interface to connect to (Ethernet or serial), selecting the device from the list of devices found and finally selecting “Get configuration from devices” and then clicking the **Connect** button. The settings from the device is retrieved and displayed within SoundStructure Studio.

Updating Firmware

After connecting to a SoundStructure device, the SoundStructure firmware may be updated using SoundStructure Studio. As the firmware files are nearly ten megabytes in size, it is recommended that SoundStructure Studio connect to the SoundStructure device over its Ethernet interface to minimize the firmware file transfer time.

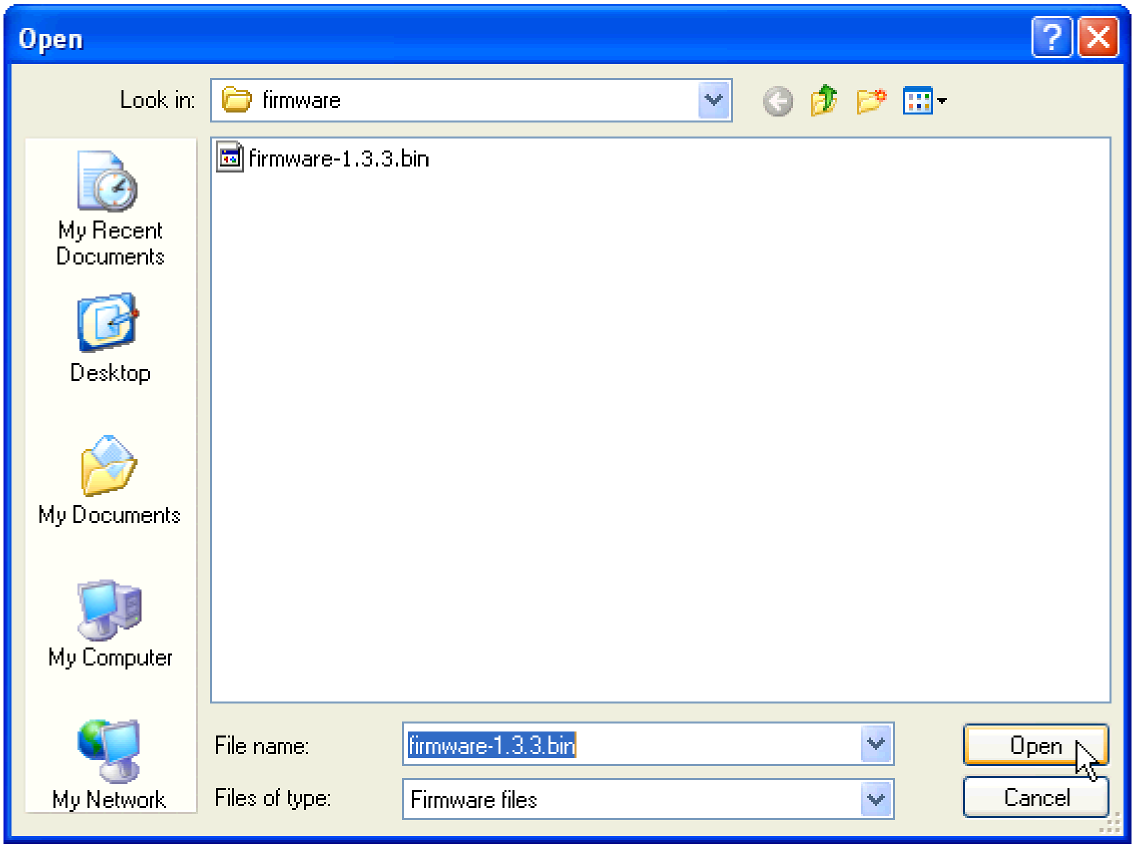
If updating firmware over RS-232, it is recommended that the 115,200 baud rate be selected on the SoundStructure device. At 115,200 baud, a typical firmware file transfer requires approximately ten to fifteen minutes. When the Ethernet interface is used, the file transfer time is reduced to less than two minutes.

After connecting to a device as described in the previous sections, click on the System name - SoundStructure System in this example - to navigate to the firmware update page shown in the following figure.


Firmware Update
Open...
Update

Click on **Open** and navigate to the directory that contains the firmware file to upload as shown in the following figure.

Firmware File

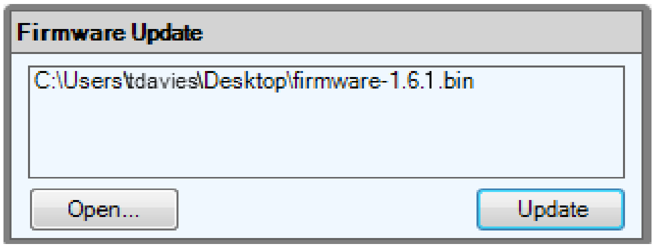

Open
Look in:
firmware
firmware-1.3.3.bin
My Recent Documents
Desktop
My Documents
My Computer
My Network
Open
Cancel

File name: firmware-1.3.3.bin

Files of type: Firmware files

Select the file by double clicking on the desired file name. Once the file has been selected, the firmware update page displays as in the following figure.

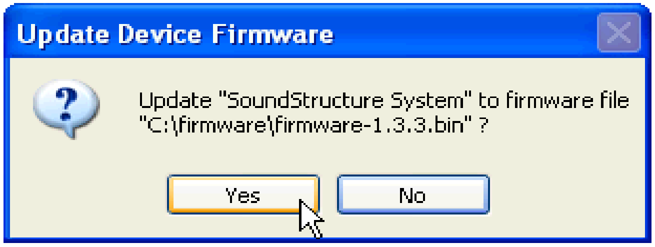
Updating Firmware



C:\Users\tdavies\Desktop\firmware-1.6.1.bin

Click on **Update** to begin the firmware transfer to the device. A window displays to confirm that the firmware file should be sent to the selected device. Select Yes to continue the firmware transfer or No to not update firmware.

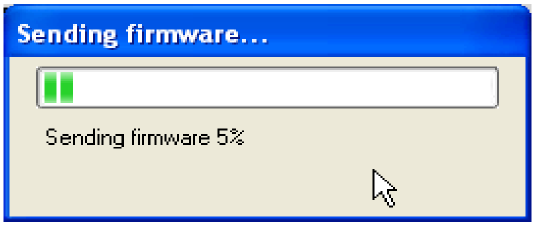
Updating Firmware Confirmation



Update "SoundStructure System" to firmware file "C:\firmware\firmware-1.3.3.bin" ?

Once the firmware transfer begins, the progress is updated with a display as shown in the following figure.

Firmware Update Progress



Sending firmware 5%

Upon completion of a successful firmware transfer, the SoundStructure device reboots and SoundStructure Studio presents the Connect to Devices window to allow SoundStructure Studio to re-connect to the device. Wait for the device to finish re-booting (front-panel green light stops flashing) and connect to the device.

If a firmware transfer is not completed successfully - perhaps because power was lost to the device or the transfer cable was mistakenly pulled out - the SoundStructure system reverts back to the firmware that was in the device prior to the firmware update process was initiated.

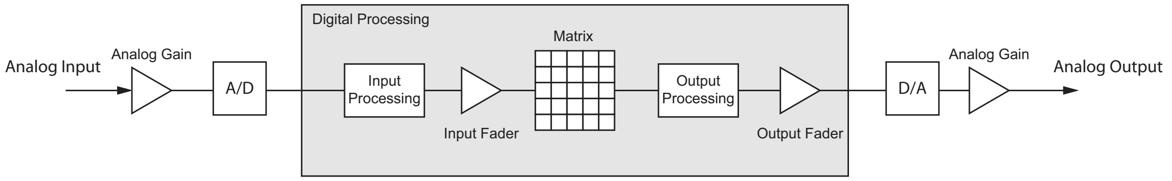
Configuring The Signal Gains

Once the SoundStructure device settings are synchronized with SoundStructure Studio, either by uploading or downloading a configuration file, the next step is to ensure the input signals have the proper analog gain to get to the 0 dBu nominal signal level of the SoundStructure devices.

SoundStructure devices may have gain applied in various positions throughout the signal chain as shown in the following figure. Gain may be applied in the analog input gain stage, the input fader, the matrix, the output fader, and the output analog gain stage.

The analog input gain is applied in the analog domain to the analog input signal to adjust the signal level to match the level required by the Analog to Digital converter to properly digitize the signal with the required signal fidelity.

Applied Analog Gain in Signal Chain

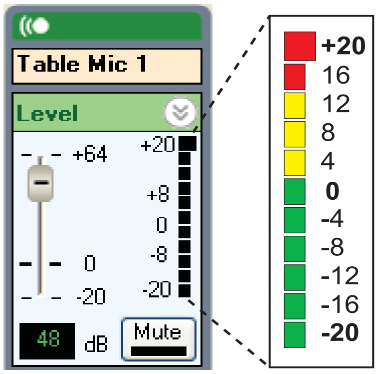


Input Signal Level Adjustment

The analog input gains are adjusted with the input gain slider on the SoundStructure Studio channels page. Any slider adjustments cause the mic_in_gain command to be executed. The analog input gain slider provides an adjustable range from -20 to +64dB of gain in 0.5dB gain steps and has a meter that shows the input signal activity from -20 to +20 dBu as shown in the following figure.

The purpose of the analog input gain is to provide enough gain to get the input signal to the 0 dBu nominal signal level of the SoundStructure devices and have additional headroom for the signal to peak above that level.

Analog Input Gain



The input signal meter is labeled so that signals greater than -20dB light the first meter segment, greater than -16dB light the second meter segment, and finally greater than +16 light the tenth meter segment. In this sense, the meter segment label represents the minimum signal level required to light the meter segment. The clip indicator at +20 illuminates when the signal exceeds +20dB.

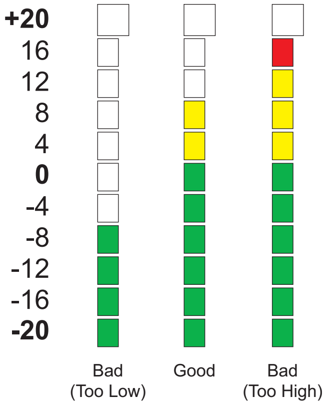
Signal Meters

The meters on the SoundStructure devices show a VU average signal level with a peak meter overlaid on the VU meter. The VU meter drives the meter segment display while the peak meter shows the maximum amplitude. The peak meters conform to the IEC 60268-18 standard and have a 12dB/second decay from the peak signal levels.

To properly adjust the levels for microphones, adjust the analog input gains so that during normal speech at the desired distance from the microphones two yellow LEDs are reached by the peak meter and occasionally additional LEDs are flickering above that. The VU meter should show a solid green LED at the 0dB level.

The following figure shows examples of peak signal levels that are too low, just right, and too high during normal conversational speech at the desired distance from the microphone.

Peak Signal Gains



If the meter levels are too low for a given microphone and for the desired distance from the microphone, increase the input gain slider to add more gain to the signal in the analog domain. As a starting point for adjusting gains, consider the following table that lists microphone sensitivities with the analog input gain required to create a 0 dBu nominal signal level in the SoundStructure products assuming a 72dB SPL audio signal at the microphone. The sensitivity information includes both dBV/Pa and mV/Pa formats and the microphone gains in this table have been rounded to the nearest 0.5dB.

SoundStructure devices provide up to 64dB of analog gain to support microphones with sensitivities as low as -44 dBV/Pa (or 6.3 mV/Pa). Microphones that have a lower sensitivity may require additional external signal gain to provide enough gain to get to the 0 dBu nominal signal level. A microphone with higher sensitivity means that less gain is required to achieve a 0dBu nominal signal when a 72dB SPL signal is present at the microphone.

For example, a common tabletop microphone has a sensitivity of -27.5 dBV/Pa. which translates to an input gain of 48dB.

Sensitivity Information for Tabletop Microphones

Sensitivity (dBV/Pa)	Microphone gain (dB)	Sensitivity (mV/Pa)
-50.0	70.0	3.2
-48.0	68.0	4.0
-46.0	66.0	5.0
-44.0	64.0	6.3
-42.0	62.0	7.9
-40.0	60.0	10.0
-38.0	58.0	12.6
-36.0	56.0	15.8
-34.0	54.0	20
-32.0	52.0	25.1

Sensitivity Information for Tabletop Microphones

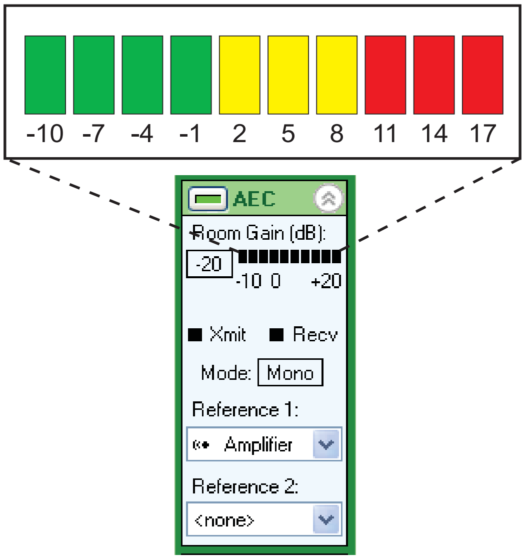
-30.0	50.0	31.6
-28.0	48.0	39.8
-26.0	46.0	50.1
-24.0	44.0	63.1
-22.0	42.0	79.4
-20.0	40.0	100.0

Room Gain

Room gain meters are used to measure the relative level of the remote audio that is present at the input to the AEC with the level of the echo that is present at the microphone. For more information on room gain and how it is measured, see Appendix C.

The room gain meter is shown on the AEC portion of the input channel on the channels page as shown in the following figure. The meter segments show the room gain ranges in 3dB increments from -10 to +20dB. The first segment of the meter is lit if the room gain is greater than -10dB and less than or equal to -7, and so on through the meter segments. The last meter segment illuminates if the room gain is greater than 17dB.

Room Gain Meter



Room Gain (dB): -20

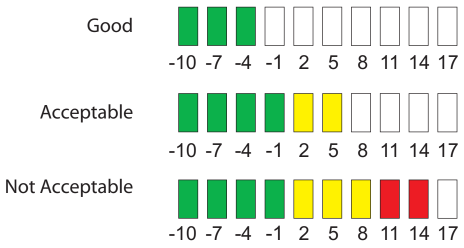
Mode: Mono

Reference 1: Amplifier

Reference 2: <none>

The following figure shows different room gain measurements that may be found in a typical room. Room gain is considered good if it is negative, meaning that the echo picked up by the microphone is less than the level that is output to the amplifier. Acceptable room gain occurs when the room gain is less than +10dB. Not acceptable room gain occurs when the room gain exceeds +10dB.

Typical Room Gain Measurements



Tabletop microphone applications typically have room gains that are 0 or less while ceiling microphone applications typically have room gains that are positive due to the proximity of the loudspeakers and ceiling microphones.

Negative room gain indicates that the AEC has a good level for the AEC reference and there is not excessive acoustic echo. Positive room gain indicates that the relative levels of the AEC reference to the microphone input should be reviewed and if the level of the reference is too low, the input gains of the remote audio sources may need to be increased while at the same time the in-room amplifier level reduced so that the overall level remains the same.

Reducing High Room Gain

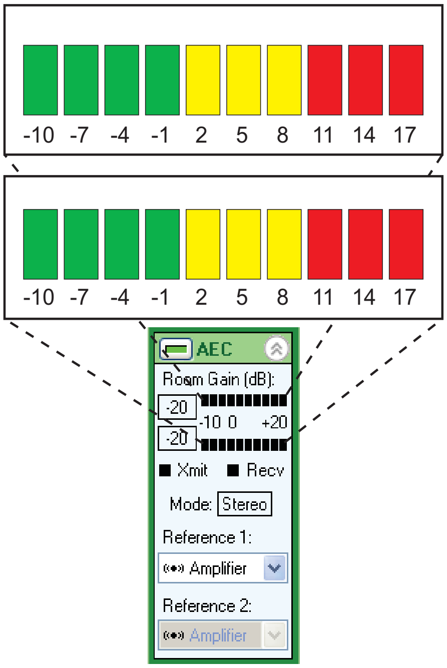
A common issue is for the AEC reference signal level, the remote audio, to be too low and the in-room amplifier turned up to compensate for the lower signal level coming into the SoundStructure device. When this happens, the room gain is increased by the amount the amplifier gain is increased. The convergence of the AEC can slow down when the room gain exceeds approximately +10dB. In general, the higher the room gain the longer it may take for the AEC to converge completely. This may have the effect of the remote site hearing residual echoes while the AEC converges.

To fix this issue, check the input signal level meters for the remote audio that is coming into the SoundStructure device to ensure that the appropriate signal gain has been applied. When the level of the remote audio is increased the in-room volume also is increased and the amplifier should be turned down to compensate for the higher signal level.

Another common issue is for the loudspeaker audio to be coupled directly into a microphone. In this case, it may be necessary to relocate the microphone away from the loudspeaker source, redirect a directional microphone away from a loudspeaker, or reduce the input gain on the microphone or amplifier to reduce the level of the echo picked up by the microphone.

When two mono AEC references are used, or a stereo virtual channel is used as the reference as shown in the following figure, there are two room gain indicators, one for each reference.

Two Room Gain Indicators



Mode: Stereo

Reference 1: Amplifier

Reference 2: Amplifier

The room gain measurements and guidelines for the two reference applications are similar to the single AEC reference example. If either reference shows a high room gain, review the gain settings for the AEC references and audio amplifier, check the microphone to loudspeaker coupling, and adjust remote audio input levels as necessary to achieve an acceptable room gain level, as described previously.

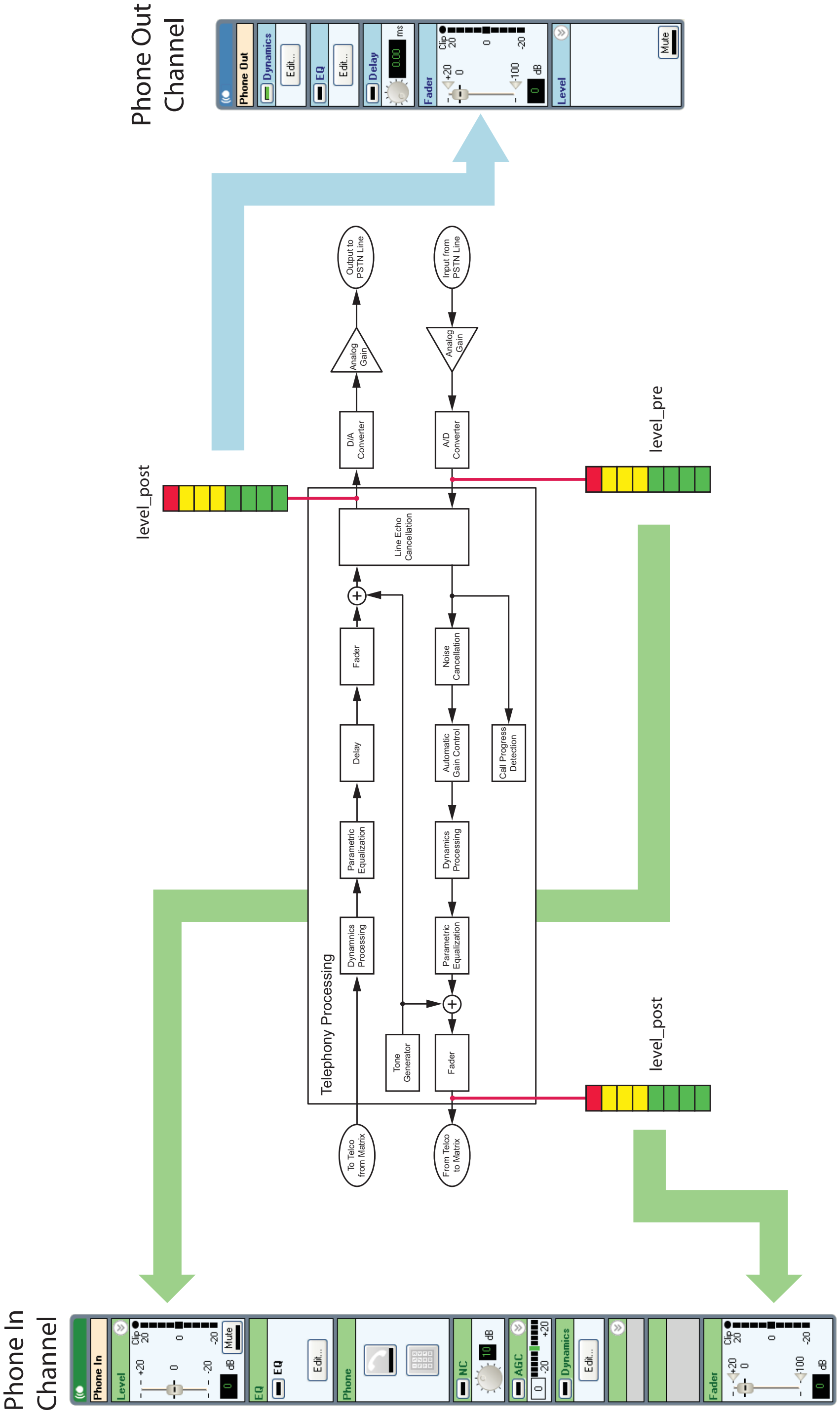
Telephony Signal Levels

The telephony inputs and outputs have an analog input gain that can be adjusted to create the required signal level on the telephone receive path. The following figure shows the input and output signal meters and where they appear within the user interface of the SoundStructure Studio software.

The Phone In gain adjusts the analog signal level coming in from the phone line. Any adjustments made to the analog input gain is reflected in the meter activity of the Phone In channel. Adjust the phone in gain so that the remote talkers peak level lights at least the second yellow LED and flickers the LEDs above that. Depending on the PBX or the Central Office connection, this could be a gain in the range of 0 to 6dB. Up to 20dB of gain may be applied at the phone input gain.

The Phone Out fader adjusts the signal level transmitted to the phone line. Any adjustments made to the output fader is reflected in the meter activity of the Phone Out channel.

Input and Output Signal Meters


Phone Out Channel
Phone Out
Dynamics
Edit...
EQ
Edit...
Delay
0.00
ms
Fader
Clip
20
0
-20
+20
0
-100
dB
0
Level
Mute
Output to PSTN Line
Input from PSTN Line
Analog Gain
Analog Gain
D/A Converter
A/D Converter
level_post
level_pre
Line Echo Cancellation
Fader
Delay
Parametric Equalization
Dynamnics Processing
Tone Generator
Noise Cancellation
Automatic Gain Control
Call Progress Detection
Dynamics Processing
Parametric Equalization
Fader
Telephony Processing
To Telco from Matrix
From Telco to Matrix
level_post
Phone In Channel
Phone In
Level
Clip
20
0
-20
+20
0
-20
dB
0
Mute
EQ
Edit...
Phone
NC
10
dB
AGC
+20
0
-20
Dynamics
Edit...
Fader
Clip
20
0
-20
+20
0
-100
dB
0

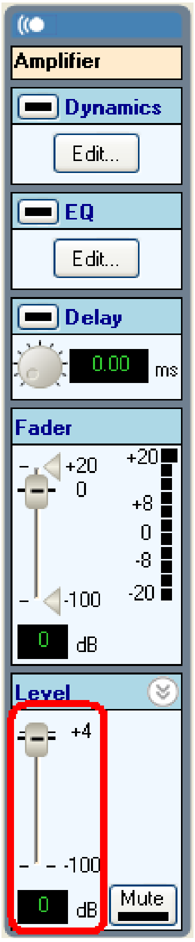
Output Signal Levels

Output signals from the SoundStructure device are connected to various other devices including audio amplifiers, recorders, and video codecs. For best performance, the output signal levels of the SoundStructure devices should match the expected signal levels of the next device that is attached.

The SoundStructure default output signal level of 0 dBu is the correct level when connecting to most professional audio equipment. When connecting to consumer equipment, such as equipment that requires an RCA-style connector the SoundStructure output gain should be reduced to -10dB to prevent overdriving the input stage on the consumer equipment. The output gain settings are found at the bottom of the channels page as shown in the following figure. The gain may be set to +4dB if required to connect to devices that require a +4dBu nominal input signal level. Negative gain adjustments (< 0) are applied in the analog domain at the digital to analog converter. Adjustments made to the output level in the highlighted slider are not shown in the fader meter.

After the output level has been set appropriately for the next piece of equipment in the signal chain, volume adjustments should be done with the fader control and adjustments in level made in the fader is shown as more or less signal in meter next to the fader control.

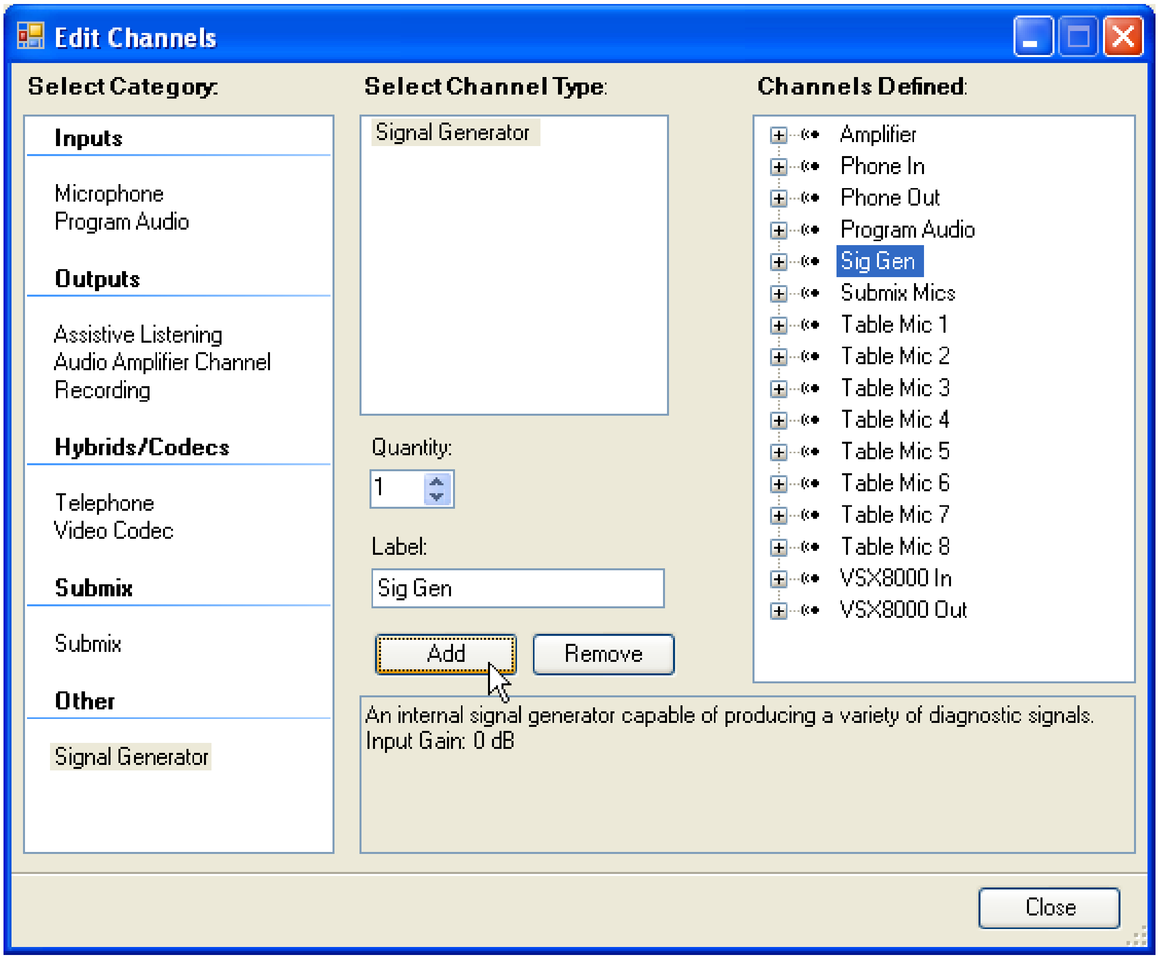
Output Gain Settings


Amplifier
Dynamics
Edit...
EQ
Edit...
Delay
0.00
ms
Fader
+20
0
-100
+20
+8
0
-8
-20
0
dB
Level
+4
-100
0
dB
Mute

Setting Amplifier Levels

It is important to set the proper level of the audio amplifier in the room. This can be done with the following steps using the SoundStructure noise generator and an SPL meter. If there are no SPL meters than can be used, the ears of the local participants can be used to help set a comfortable level in the room.

1 If there isn’t already a signal generator as part of the project, add a signal generator to the project by selecting Edit Channels and select the Signal Generator as shown in the following figure.

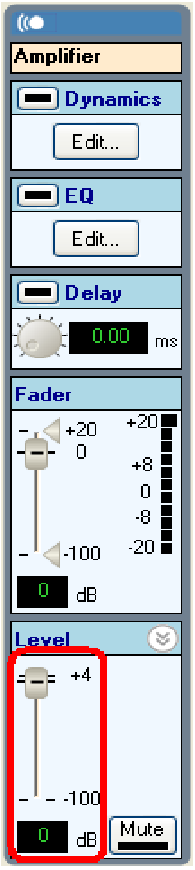

Edit Channels
Select Category:
Inputs
Microphone
Program Audio
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Telephone
Video Codec
Submix
Submix
Other
Signal Generator
Select Channel Type:
Signal Generator
Channels Defined:
Amplifier
Phone In
Phone Out
Program Audio
Sig Gen
Submix Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
VSX8000 In
VSX8000 Out
Add
Remove
Close

Quantity: 1

Label: Sig Gen

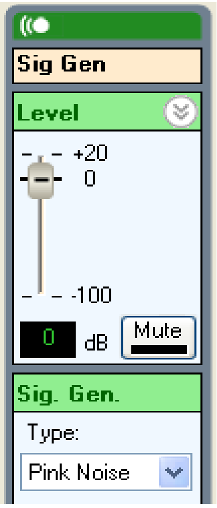
An internal signal generator capable of producing a variety of diagnostic signals.
Input Gain: 0 dB

2 Set the analog output gain on the amplifier output channel to be either +4, 0, or -10 depending on the nominal signal level required by the audio amplifier. Amplifiers with RCA inputs require a -10dB setting, most system integration professional amplifiers require the 0dB setting, and some amplifiers require the +4dB setting.


Amplifier
Dynamics
Edit...
EQ
Edit...
Delay
0.00
ms
Fader
+20
0
-100
+20
+8
0
-8
-20
0
dB
Level
+4
-100
0
dB
Mute

3 Turn down the audio amplifier to the *lowest* possible volume setting (alternatively the highest amount of attenuation). The noise generator is loud in the next step and it is best to reduce the gain on the amplifier prior to sending noise into the room.

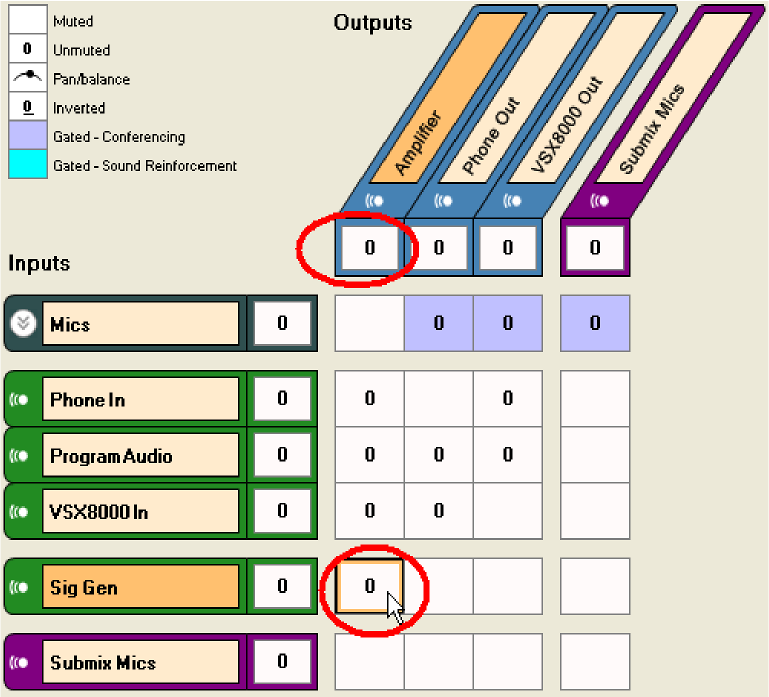
4 On the channels page unmute the signal generator and ensure the gain is set to 0 as shown in the following figure. There are different signals that the signal generator can create, ensure that Pink Noise is selected.



0 dB

Type: Pink Noise

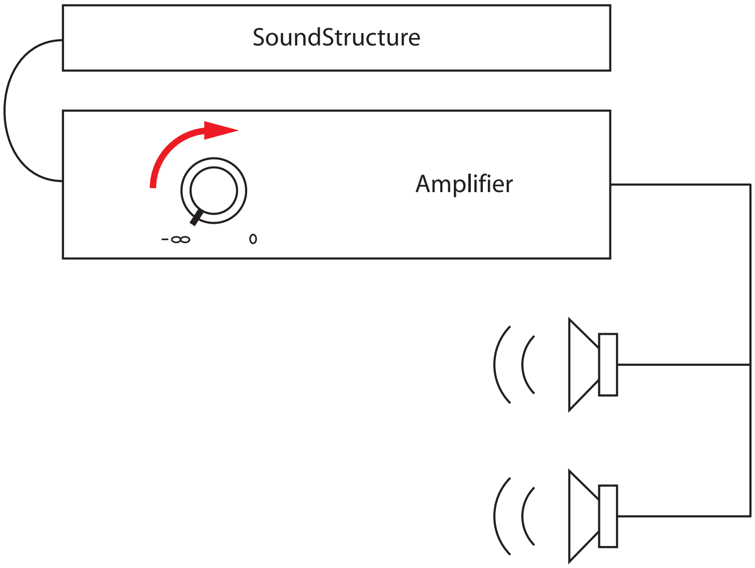
5 Set the output fader from the SoundStructure device to 0 as shown in the next figure and unmute the signal generator to the loudspeaker output. Pink noise may be heard in the room depending on the amplifier volume settings.



	Muted
0	Unmuted
⌒	Pan/balance
0	Inverted
🟪	Gated - Conferencing
🟦	Gated - Sound Reinforcement

Inputs		Amplifier	Phone Out	VSX8000 Out	Submix Mics
Outputs		0	0	0	0
Mics	0		0	0	0
Phone In	0	0		0	
Program Audio	0	0	0	0	
VSX8000 In	0	0	0		
Sig Gen	0	0			
Submix Mics	0				

6 Adjust the audio amplifier volume knob until the SPL meter, positioned at the listener's ear position, measures approximately 75dB SPL C-weighted.



The target level of 75dB SPL is selected because pink noise is approximately 10 to 12dB louder than human speech. Adjusting the amplifier volume so that 75dB SPL is heard in the room ensures that when speech is played into the room the speech is at a good level for the listeners. Additional volume control can be performed by adjusting the level of the fader on the “Amplifier” channel within the SoundStructure device. Creating Advanced Applications provides examples of using the “Amplifier” channel for volume control.

Presets

Once any settings of the SoundStructure system have been adjusted, it is important to save the settings to a full preset to ensure the settings survive a power cycle.

There are two types of presets supported within SoundStructure systems - full presets and partial presets. All presets are stored as part of the SoundStructure configuration file.

Full Presets

Full presets store all the audio parameters of the virtual channels including input and output gains, signal processing options, matrix cross point settings, automixer settings, and all other signal-related settings that are different from the default values for these parameters.

SoundStructure presets do *not* store device-level information such as the RS-232 rate, Ethernet address, virtual channel definitions, virtual channel group definitions, or logic pin definitions. These settings are defined in a separate section of the configuration file and can not be changed as part of preset execution.

When full presets are executed there are two distinct events that happen. First the default values for all parameters are restored and then the full preset is restored. The analog outputs of the system are muted during the time it takes to execute the full preset.

Note: Analog Outputs Muted During Full Preset Execution

The analog outputs of the SoundStructure system are muted during the time it takes to execute a full preset.

Partial Presets

Partial presets store only the settings that a user places into the partial preset. Partial presets are designed for use with volume control applications and muting multiple signals and any other applications where it is necessary to run multiple commands with a single API command. Any parameter within SoundStructure may be adjusted with a partial preset. Partial presets are similar to the concept of macros in the Polycom Vortex products.

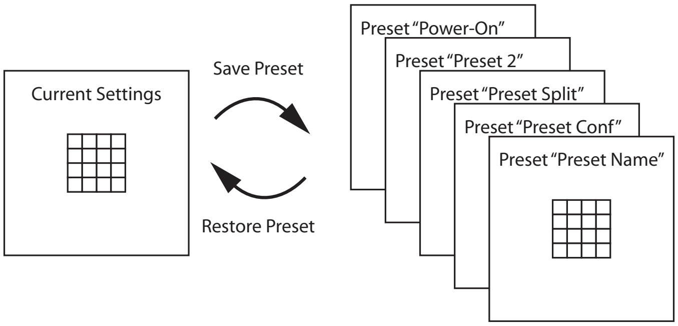
When partial presets are executed, the commands in the partial presets are executed sequentially with the first command listed executing first.

Preset Operation

SoundStructure devices store presets in non-volatile memory to ensure the preset settings are not lost upon power cycling. When presets are executed, all the parameter settings for the preset are copied into the current device settings which are stored in RAM and become the parameters the device operates from. Any adjustments to the device settings, such as volume adjustments or muting, make adjustments to the RAM-based current settings of the device. When the current settings are saved to a preset, the current settings are stored to the non-volatile memory with a default preset name. The preset name may be customized as described next.

Unless the current settings are copied to a preset using the Preset Save function, the current settings are be lost upon power cycling. Using SoundStructure Studio, current settings of the device can be saved to full presets and restored from full presets as shown in the following figure.

Saving and Restoring Full Presets



Power-On Full Preset

SoundStructure full presets operate in a similar fashion to Polycom's Vortex products where there is a "power on" preset that must be selected for the design when the device powers up. When creating a new design, the last step of the SoundStructure design process saves the settings to a preset called "Power-On" and sets the power on preset to that preset.

When a SoundStructure device boots up, it reads its internal configuration file and defines its virtual channels and virtual channel groups, sets the system default values for these channels and groups, and then looks for the power on preset. If the power on preset is found, the system boots to the power on preset. If the power on preset is not found, any current settings that may have been stored in the configuration file is restored. If the current settings are not found, then the factory default settings are used. Note that the factory default settings are not necessarily be useful as matrix cross points are muted by default and gains are set to 0dB.

Preset Names

When presets are stored, the preset name may be customized to any arbitrary string of up to 256 bytes in length. When naming presets, keep in mind the preset name is used in the command syntax to invoke the execution of the preset. It is recommended that a preset name be descriptive to aid in selecting presets for execution from within SoundStructure Studio. As described later in this chapter, presets are executed with the run command with the preset name as an argument.

Number Of Presets

The number of presets is limited only by the amount of available non-volatile memory in the SoundStructure system. For single device installations, it is estimated that more than 100 full presets may be stored in the device.

Saving Presets

After the system has been designed with SoundStructure Studio, there is a default full preset called “Power-On” and the preset is assigned to be the power-on preset. If any changes are made to virtual channel parameters or matrix cross points are adjusted, the updated settings should be saved in the “Power-On” preset by selecting the **Save Selected** preset as shown in the following figure.

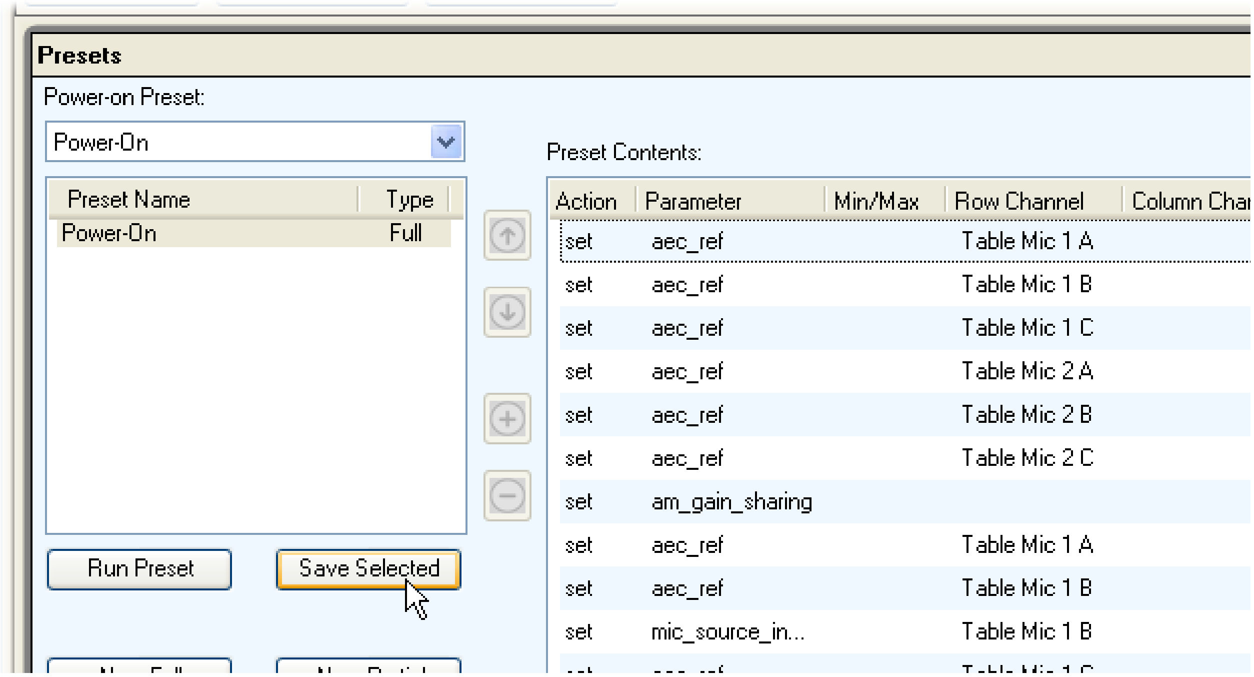
When a preset is saved, all the audio settings of the device are compared to their factory default settings, and only the settings that differ from the default settings are stored in the preset. By comparing presets to a default set of values, the size of the presets are reduced which allows more presets to be stored in the device.

Note: Changes to Default Settings Stored in Full Presets

Full presets store the differences from the default settings. If a parameter isn’t shown in a full preset, it is because that value of that parameter is the same as the default value.

The preset page shows the presets and also the preset contents to make it possible to determine the settings that are in each preset. The column headings may be selected to sort the preset based on the values in the column. Changing sort order does not change the order of execution if the entries are in the preset.

Save Selected Preset



Action	Parameter	Min/Max	Row Channel
set	aec_ref		Table Mic 1 A
set	aec_ref		Table Mic 1 B
set	aec_ref		Table Mic 1 C
set	aec_ref		Table Mic 2 A
set	aec_ref		Table Mic 2 B
set	aec_ref		Table Mic 2 C
set	am_gain_sharing		
set	aec_ref		Table Mic 1 A
set	aec_ref		Table Mic 1 B
set	mic_source_in...		Table Mic 1 B

The column headers of the preset information are shown in the following table.

Preset Information Column Headers

Column Header	Description
Action	The action that is applied to the parameter. Typically the action is **set** for full presets although for partial presets the action could be set, inc, dec depending on the desired behavior in the partial preset. See Appendix A for the description of the actions and how they are used with the API.
Parameter	This is the parameter that is adjusted when the preset executes. Examples of parameters include mute, fader, aec_en, etc.
Min/Max	This is how the minimum or maximum value of a parameter, such as a fader, can be adjusted.
Row Channel	This is the virtual channel name who’s parameter is being adjusted.
Col Channel	For parameters that affect matrix crosspoints, this is the name of the output virtual channel.
Index	This is the way to get access to the individual parameters that if multiple parameters are associated with a parameter such as the AEC reference.
Value	This is the value that the action applies to the parameter of the Row Channel or at the Row Channel and Col Channel.

Saving a preset to the SoundStructure system causes the preset to be written into the non-volatile memory of the SoundStructure device. When online, the settings are transferred to the SoundStructure device and stored in the non-volatile memory.

The current settings of a device may be saved to a new *full* preset by selecting the **New Full** preset option. The new preset has a default name of "New Preset" and the name can be changed by left clicking on the preset name.

Presets may be saved, removed, or re-named only from within the SoundStructure Studio software. Presets may be executed via the SoundStructure API as described next by using the *run* action.

Virtual Channels And Groups And Presets

Full presets store all the parameter settings that differ from the defaults for all the virtual channels that are defined at the time the full preset is created.

If, after a full preset has been saved, a new virtual channel is defined or renamed, the existing presets are updated with the new channel name at the time that any full preset is saved, any full preset is executed from SoundStructure Studio, or the configuration file is saved using the File Save option.

If virtual channels are removed, then all presets that have any reference to that virtual channel is updated when any preset is saved, any preset is executed from SoundStructure Studio, or the configuration file is saved using the File Save option.

Note: Saving Current Settings to a Preset
Any changes to current settings that are desired to survive a power cycle must be saved to a preset, and usually the power-on preset, if the settings are to survive a power cycling.

Creating Partial Presets

Partial presets are a list of commands that are executed when the partial preset is run. Partial presets can be created in three ways:

- Removing entries from a full preset
- Creating new blank partial presets
- Using the preset recording tool

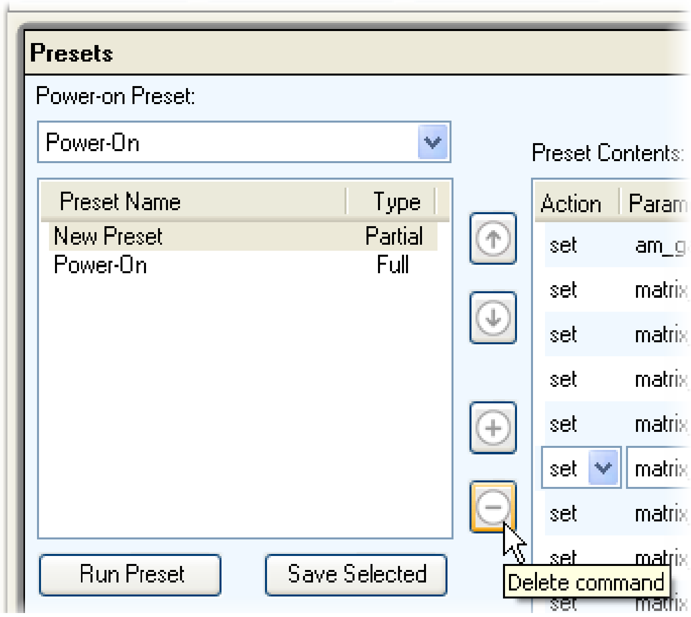
Creating a Partial Preset from a Full Preset

Partial presets consist of a sequence of commands that are executed in the order they appear in the partial preset. If an entry is removed from a full preset, the full preset becomes a partial preset.

If there is only one full preset, entries in the preset may not be deleted or added to ensure there is at least one full preset.

When there is more than one full preset, entries in a preset may be removed by clicking the '-' symbol as shown in the following figure. Once a line is deleted from a full preset, it becomes a partial preset automatically.

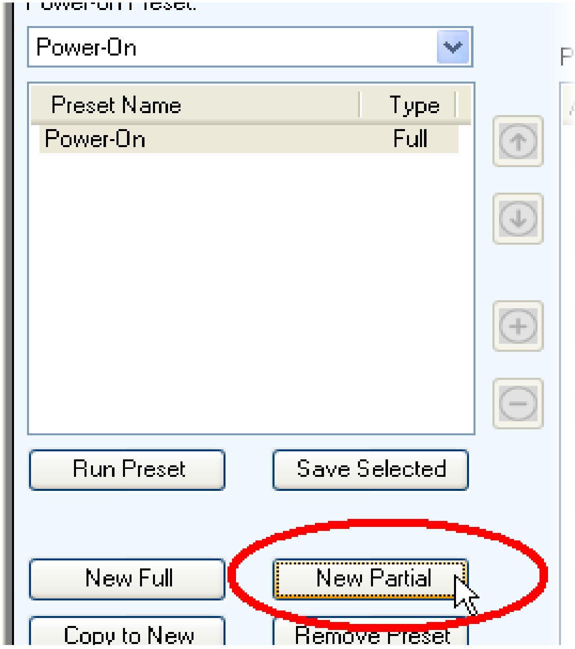
Removing Presets


Presets
Power-on Preset:
Power-On
Preset Name
Type
New Preset
Partial
Power-On
Full
Preset Contents:
Action
Param
set
am_g
set
matrix
Delete command
Run Preset
Save Selected

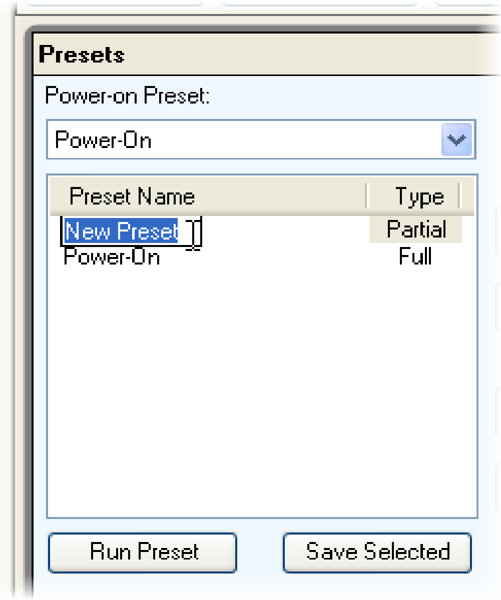
Creating Partial Presets Manually

New partial presets may be created by selecting 'New Partial' as shown in the following figure.

Creating Presets


Power-On
Preset Name
Type
Power-On
Full
Run Preset
Save Selected
New Full
New Partial
Copy to New
Remove Preset

The next step is to enter a name for the partial preset as shown in the following figure.

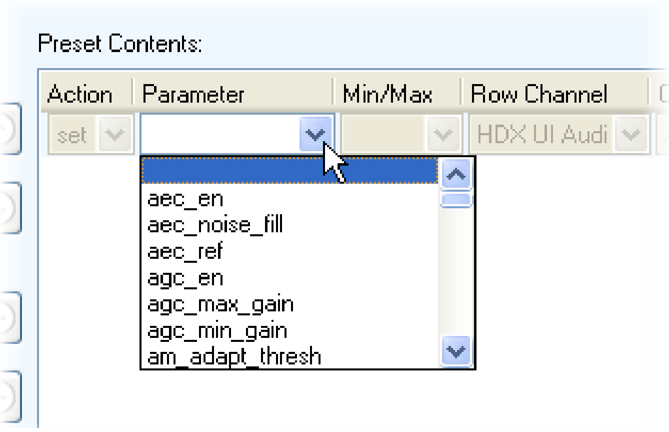


Power-on Preset: Power-On

Preset Name	Type
New Preset	Partial
Power-On	Full

After the empty partial preset has been created, the next step is to add commands to the partial preset by clicking the '+' control. This adds an empty line to the partial preset, and allow the designer to select the parameter to adjust with this line as shown in the following figure.

Selecting Parameters



Preset Contents:

Action	Parameter	Min/Max	Row Channel
set			HDX UI Audi

Partial presets are entered one command at a time by pulling down the appropriate parameter and adjusting the action (set, inc, dec, tog) and selecting the arguments for the parameter.

The column headers of the partial preset may not be sorted as this would change the execution order of the partial preset. However the column widths may be adjusted on the preset content table to allow showing the full parameters that are being added.

Once the contents have been added to the preset, ensure the ‘Save Selected’ button is pressed to ensure the preset contents are saved with the configuration file. Navigating away from the partial preset prior to saving removes all the entries from the partial preset.

Note: Saving Partial Presets

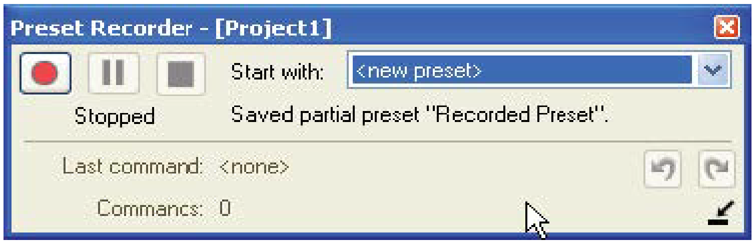
To save the partial preset after adding commands, select the ‘Save Selected’ button.

Creating Partial Presets with the Preset Recorder

You may record partial presets with the partial preset recorder tool found under the Tools Menu and with the Record Preset button on the Preset page.

The recorder, as shown below, has the following features listed in the table below.

Preset Recorder



Start with: <new preset>

Stopped

Saved partial preset "Recorded Preset".

Last command: <none>

Commands: 0

Recorder Control Features

Control	Description
Start Recording	All UI interaction after starting the recording is stored to a partial preset.
Pause Recording	Temporarily stops recording commands.
Stop Recording	Finishes the partial preset recording.
Starting Preset	You can either record a new preset or append to an existing preset depending on the value of this field
Redo Last Command	Redoes a command that was undone with the Undo button.
Undo Last Command	Undoes the last command recorded so it is not part of the preset.
Last Command Recorded	Shows the last command that was recorded.
Number of Commands	Shows the number of commands that have been recorded with the partial preset.

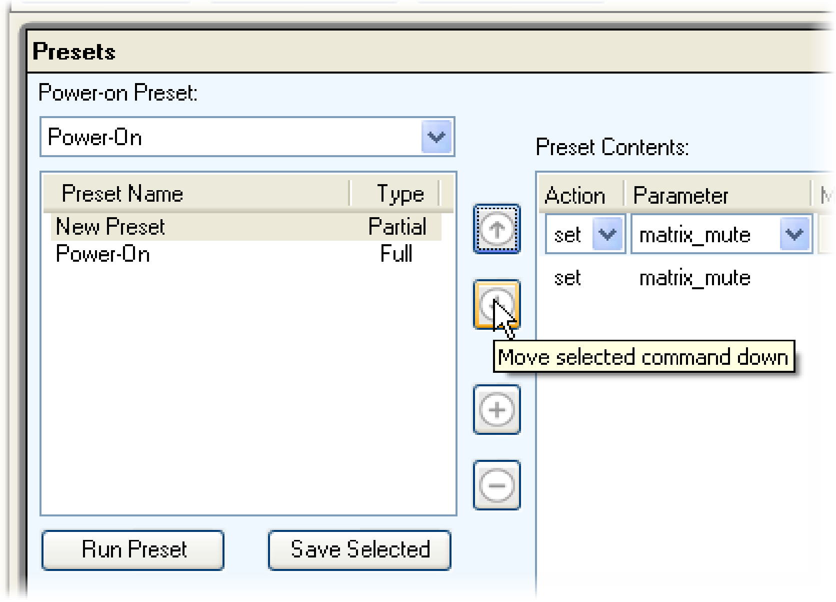
Once recorded, the presets are available for review and manual editing in the Presets page.

Reordering Entries In A Partial Preset

The order of execution of the commands in a partial preset may be adjusted by changing the order of the lines in the partial preset. To move a line, select the line and select the up or down arrow as shown in the following figure.

After entries have been moved, select **Save Selected** to save the new execution order.

Saving Presets


Presets
Power-on Preset:
Power-On
Preset Name
Type
New Preset
Partial
Power-On
Full
Preset Contents:
Action
Parameter
set
matrix_mute
set
matrix_mute
Move selected command down
Run Preset
Save Selected

Running Presets

Both full and partial presets may be executed when in SoundStructure Studio by left clicking the preset to execute and then clicking **Run Preset**.

A control system would execute the preset with the command action *run* as in the following example:

```
run “Power-On”
```

In version 1.5 or later firmware, once a preset run command has been executed, the SoundStructure system immediately responds with an acknowledgment as shown below to indicate that it is running a preset.

```
run “Power-On”
```

Full Presets

When a full preset is executed all the outputs of the system are muted during the execution of the preset and then unmuted after the full preset finishes executing.

No command acknowledgments are generated when a full preset is executed. If there are parameters that a control system needs to know the value of, these parameters should be queried after the execution of the preset.

Once the full preset has finished running, a ‘ran’ command acknowledgment is generated as shown next.

```
ran “Power-On”
```

Partial Presets

Partial presets generate command acknowledgments for all parameters that are changed during the execution of the preset. The outputs of the system are ***not*** muted during a partial preset unless the designer explicitly inserts commands to mute the outputs of the system during the partial preset.

Once the full preset has finished running, a ‘ran’ command acknowledgment is generated as shown next where “Partial Preset” would be replaced with the name of the partial preset that was executed.

```
ran “Partial Preset”
```

Full preset execution does not generate any command acknowledgments from the SoundStructure system. If specific parameters are required after a preset has been executed, the values for the parameters should be queried after a preset has executed.

The outputs of the system are muted during the execution of a full preset. The outputs are unmuted after the preset has executed. This muting does not affect the state of the safety mute or any other mute parameter.

Removing Presets

Presets may be removed from the system by left clicking on the preset and then clicking **Remove Preset**.

If the power-on preset is removed and the system rebooted, the system boots into the current settings if they have been stored in the configuration file.

Using Events, Logic, and IR

This chapter introduces the new concept of Events and how events may be created and used to control the behavior of SoundStructure systems including using the logic input and output capabilities.

Understanding Events

SoundStructure Studio and firmware version 1.3 introduces the concept of Events. Events are built from sources, triggers, and actions as described in the following sections. Events are used to connect both external stimuli, such as logic input pins or IR key presses, and internal stimuli such as mute and gating status, to control settings within SoundStructure such as executing presets, or muting microphones. Events are also used to integrate the Polycom IR remote controller with a SoundStructure device, allowing the different key presses to execute functions within SoundStructure such as taking the PSTN interface offhook, dialing digits, or muting microphones.

Sources

With an event the source defines the set of parameters that can be used to make something happen within SoundStructure. Sources may be button pushes, IR key presses, or particular SoundStructure parameters. The event sources that are allowed within SoundStructure are shown in the following table.

SoundStructure Event Sources

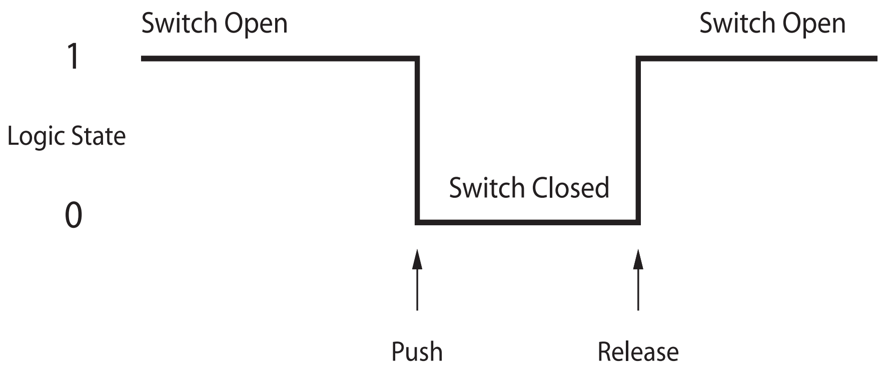
Event Sources (parameter name)	
Safety mute (safety_mute)	Codec mute status (clink_mute)
Codec volume status (clink_volume)	Temperature status (dev_temp_status)
Mute state (mute)	Call Status (clink_call_active)
Gating status (am_gate)	Fader value (fader)
Signal activity status (signal_active)	Camera gating status (am_camera_gate)
Phone ring (phone_ring)	Phone hook status (phone_connect)
Digital logic inputs held (digital_gpio_held)	Digital logic inputs (digital_gpio_state)
IR keypress (ir_key_press)	Analog logic input values (digital_gpio_value)
IR key held (ir_key_held)	

Triggers

- The trigger determines when the information in the source becomes actionable. Triggers may be one of three values: always, equals, or range as defined below.
- The **always** trigger means that **any changes** to the source parameter causes the action to execute.
- The **equals** trigger means that when the source parameter equals the desired value (e.g., open or closed for logic inputs), the action executes.
- The **range** trigger means that anytime the source parameter value is equal to or greater than the min value and less than or equal to the max value, the action executes.

Triggers are ‘edge’ triggered, meaning that when the source parameter value changes, the event engine determines whether the trigger condition is met or not. For example, for logic input switches this means there are two edges – when a logic input is closed (has a value of 0), or when a logic input is opened (has a value of 1). The following figure shows the two signal edges associated with a button press – the transition from open to closed and from closed to open. Either edge, or both edges, may be used to trigger events.

Signal Edges Associated with Button Presses



Actions

The action specifies what happens when the trigger condition is met. Actions include

- running a particular command,
- running a preset, or
- mapping the value of the source parameter to a destination parameter.

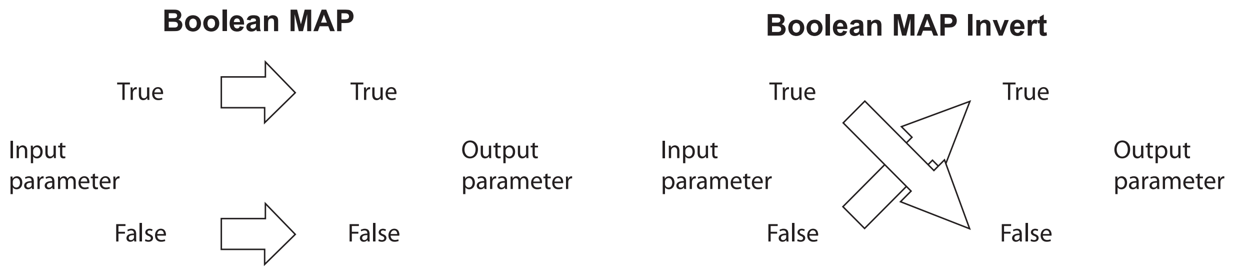
Running a command allows the event to directly execute a single command to change a SoundStructure parameter via a valid API command. An example is muting all microphones as the action for when a button is pressed.

Running a preset allows the event to execute either a partial or full preset. An example of this type of action is changing the room matrix routing when a button is pressed.

When the Always trigger is used, it is convenient to use the Map action. The map action allows the value of the destination parameter to track the value of the source parameter with a single event. Both Boolean parameters (with values of only 0 or 1, e.g., mute state) and numeric parameters (with values from a min to a max value, e.g., fader value) may be mapped.

Action maps may also be inverted to allow for the case where it is desired for the destination parameter map to be the inverse of the source parameter. An example of mapping Boolean parameters is shown in the following figure with both a direct mapping (true ◊ true) and an inverted mapping (true ◊ false).

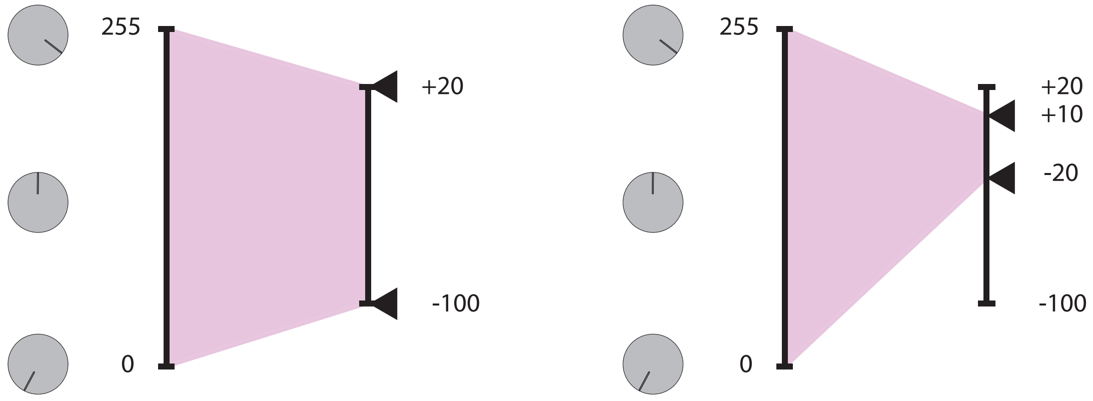
Mapping Boolean Parameters


Boolean MAP
True
True
Input parameter
Output parameter
False
False
Boolean MAP Invert
True
True
Input parameter
Output parameter
False
False

An example of mapping a Boolean parameter would be tying the mute state of a microphone to the logic output state that drives an LED. Changes in the mute state of a microphone would be mapped directly to a logic output pin.

Numeric parameters can also be used in action maps. An example of mapping a numeric parameter is mapping the analog voltage from the logic inputs to the fader of a particular channel. In the example shown in the following figure (left), the values from the volume knob are mapped to a fader. The analog voltage values from the volume knob range from 0 to 255 and are directly mapped to the fader range of -100 to +20dB. Changes in the analog input voltage map directly to the fader values in a linear fashion.

Numeric Mapping Parameters


255
0
+20
-100
255
0
+20
+10
-20
-100

In addition, if the user min and max values of the fader channel are used, the action map automatically uses the user min and max range instead of the full -100 to +20 dB range. Using a min/max range on the output parameter is shown on the right side of the figure where the user min has been set to -20 and the user max has been set to +10.

Creating Events With SoundStructure Studio

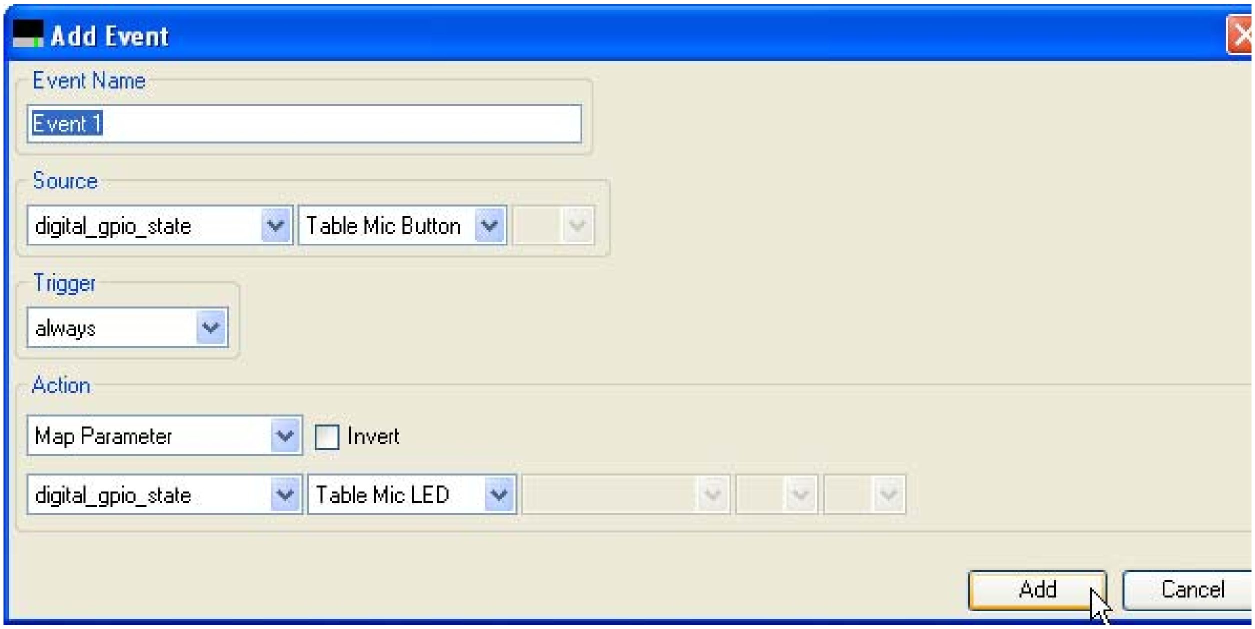
SoundStructure Studio allows the A/V designer to create events with the Add Event button on the Events page. As we'll see shortly there are a number of events that are created automatically.

Adding New Events

Clicking the add event button shows a user control that allows the designer to create an event name and then the source, trigger, and action. Similar to channel names, event names must be unique and are case sensitive. We recommend you use a name that makes sense to you.

To create an event, select the source, the trigger, and the type of action and select **Add**.

Adding an Event



Event Name: Event 1

Source: digital_gpio_state, Table Mic Button

Trigger: always

Action: Map Parameter ☐ Invert

digital_gpio_state, Table Mic LED

Once the events are created, the events page shows the entire list of events as shown in the following figure. The events page may be sorted by any of the column headings including the Event Name, Source, Parameter, Trigger, and Action.

Events may be edited by double clicking on an event on the Events page. Once completed, click Save to save the event or cancel to not save the event.

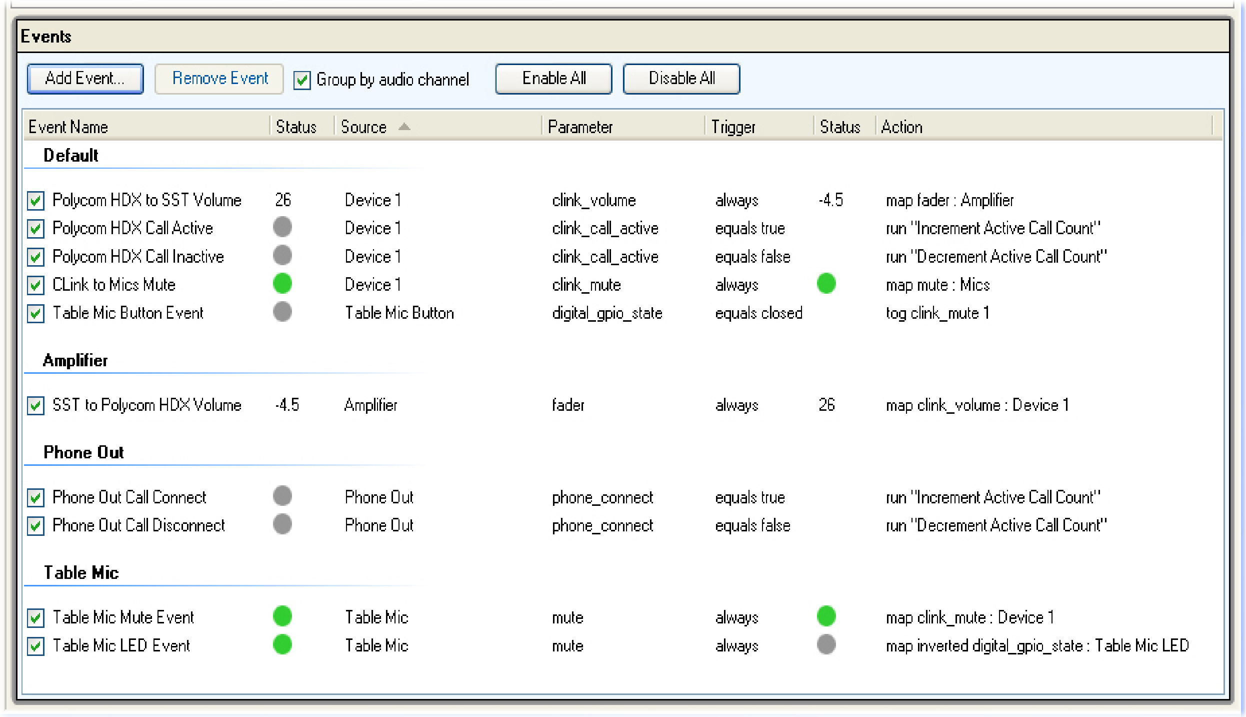
Enable And Disable Events

Events may be enabled or disabled by checking or unchecking, respectively, the box to the left of the Event name. Disabling events is intended to simplify troubleshooting a system that may have many events. Once events are enabled, the trigger is re-evaluated for all map actions and the resulting action executed.

The Enable All and Disable All buttons enable or disable all events that have been defined.

To permanently store the enable/disable state of an event, save the configuration file to disk using File Save or File Save As. When connected online to a SoundStructure system, the File Save or File Save as forces the settings in the device to be written to the flash memory of the device.

Enabling Events



Event Name	Status	Source	Parameter	Trigger	Status	Action
Default						
☑ Polycom HDX to SST Volume	26	Device 1	clink_volume	always	-4.5	map fader : Amplifier
☑ Polycom HDX Call Active	●	Device 1	clink_call_active	equals true		run "Increment Active Call Count"
☑ Polycom HDX Call Inactive	●	Device 1	clink_call_active	equals false		run "Decrement Active Call Count"
☑ CLink to Mics Mute	🟢	Device 1	clink_mute	always	🟢	map mute : Mics
☑ Table Mic Button Event	●	Table Mic Button	digital_gpio_state	equals closed		tog clink_mute 1
Amplifier						
☑ SST to Polycom HDX Volume	-4.5	Amplifier	fader	always	26	map clink_volume : Device 1
Phone Out						
☑ Phone Out Call Connect	●	Phone Out	phone_connect	equals true		run "Increment Active Call Count"
☑ Phone Out Call Disconnect	●	Phone Out	phone_connect	equals false		run "Decrement Active Call Count"
Table Mic						
☑ Table Mic Mute Event	🟢	Table Mic	mute	always	🟢	map clink_mute : Device 1
☑ Table Mic LED Event	🟢	Table Mic	mute	always	●	map inverted digital_gpio_state : Table Mic LED

Event Entries In The Logs

The SoundStructure device logs record which events executed and the resulting command acknowledgments there were generated. An example log file is shown below where the events that executed are highlighted. If in doubt whether events are executing, check the logs within a SoundStructure system.

```
Oct 10 23:05:18 gcp: cmd: [1:6:172.25.240.31] set mute "Table Mic" 1
Oct 10 23:05:18 gcp: sts: event "Table Mic LED Event" triggered
Oct 10 23:05:18 gcp: ack: [all] val digital_gpio_state "Table Mic LED" 0
Oct 10 23:05:18 gcp: sts: event "Table Mic Mute Event" triggered
Oct 10 23:05:18 gcp: sts: event "CLink to Mics Mute" triggered
Oct 10 23:05:18 gcp: ack: [all] val mute "Lectern Mic" 1
Oct 10 23:05:18 gcp: sts: event "Table Mic A Mute Event" triggered
Oct 10 23:05:18 gcp: ack: [all] val mute "Table Mic A" 1
Oct 10 23:05:18 gcp: sts: event "Table Mic B Mute Event" triggered
Oct 10 23:05:18 gcp: ack: [all] val mute "Table Mic B" 1
Oct 10 23:05:18 gcp: sts: event "Table Mic C Mute Event" triggered
Oct 10 23:05:18 gcp: ack: [all] val mute "Table Mic C" 1
Oct 10 23:05:18 gcp: ack: [all] val clink_mute 1 1
Oct 10 23:05:18 gcp: ack: [all] val mute "Table Mic" 1
```

Removing Events With Studio

Events may be removed by selecting one or more events and choosing the Remove Event button on the events page.

SoundStructure Studio Automatically Creates Events

When a new project is created, SoundStructure Studio automatically creates events depending on the input and output options selected. For example,

- If a Polycom Video Codec is added to a project, new events for volume control and mute are created. These new events replace the pre-1.3 requirement to use the virtual channel names “Mics” and “Amplifier”.
- If PSTN interfaces are added to a project, events for tracking the call active state (onhook/offhook) are created.
- If push-to-talk microphones are added, events for muting the microphones and for illuminating status LED’s are created.
- If an Polycom IR remote is selected, events mapping the key presses on the IR remote to the appropriate functions within SoundStructure are created.

Backwards Compatibility with Earlier SoundStructure Firmware

As described in this section, SoundStructure device firmware 1.3 uses events to handle the muting and volume control integration between a Polycom Video Codec and a SoundStructure device. In pre-1.3 versions of SoundStructure firmware, the names “Mics” and “Amplifier” were required for integrating the Codec Mute and Volume control to SoundStructure devices. If those names were not defined in the SoundStructure device, the Polycom Video Codec was not able to control the SoundStructure device.

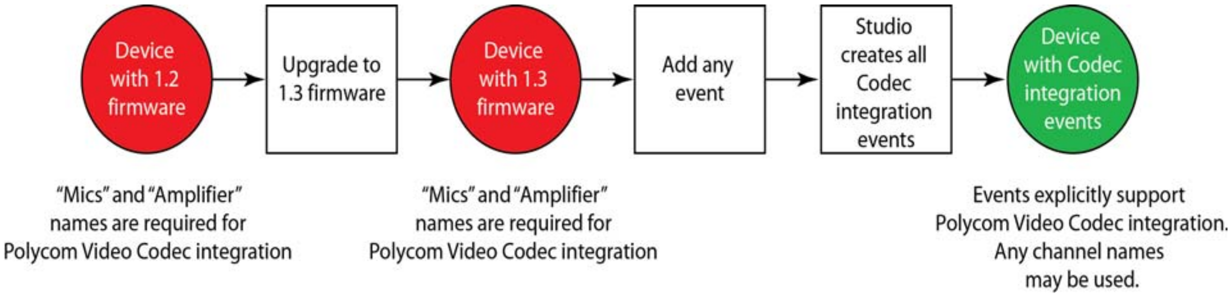
SoundStructure device version 1.3 firmware uses a different and better way to handle the codec muting and volume control.

1. If there are no events defined for the SoundStructure device, the earlier firmware behavior is retained and the names “Mics” and “Amplifier” are required for the Polycom Video Codec to control the SoundStructure device. This backward compatibility mode means that the SoundStructure system behaves as it did prior to the firmware upgrade when there were no events defined.
2. When the SoundStructure device is upgraded to the 1.3 or later firmware, defining any events using the Add Events feature causes SoundStructure Studio to automatically create the necessary SoundStructure events for the Polycom Video Codec and SoundStructure device integration.

The following figure shows that the “Mics” and “Amplifier” names from earlier firmware are required until any event is added. At that point, SoundStructure Studio creates all the necessary events to support the integration with events.

Once the new events are created, the new events are used for the integration and the name “Mics” and “Amplifier” are no longer required for Polycom Video Codec integration.

Names of Microphones and Amplifiers


Device with 1.2 firmware
Upgrade to 1.3 firmware
Device with 1.3 firmware
Add any event
Studio creates all Codec integration events
Device with Codec integration events
"Mics" and "Amplifier" names are required for Polycom Video Codec integration
"Mics" and "Amplifier" names are required for Polycom Video Codec integration
Events explicitly support Polycom Video Codec integration. Any channel names may be used.

Polycom Video Codec Integration Events

The following events are automatically generated when a Polycom Video Codec is designed as part of a SoundStructure project.

Automatically Generated Events

Event Name	Description
Polycom Video Codec to SoundStructure Volume	Maps user volume adjustments from the Polycom Video Codec (from Polycom IR key presses or from the Codec being controlled externally) to the fader on the channel "Amplifier"
SoundStructure to Polycom Video Codec Volume	Maps any user fader adjustments on the first SoundStructure amplifier channel to the Codec volume
Clink to Mics Mute	Maps the mute of the Polycom Video Codec to mute microphones on SoundStructure

Note: Maintaining Backwards Compatibility

Please note that to maintain backwards compatibility with the earlier versions of SoundStructure firmware and Polycom Video Codec integration, the "Amplifier" fader channel now has user min and max values set to -31 and +20 respectively. This means that the fader on the "Amplifier" channel does not go lower than -31dB without changing the fader user min value to a lower value.

The following Polycom Video Codec specific events are created when there is one or more SoundStructure telephony interfaces (TEL1 or TEL2)

.

Polycom Video Codec Events for Telephony Interfaces

Event Name	Description

Polycom Video Codec Events for Telephony Interfaces

Polycom Video Codec Call Active	Increments the local `clink_local_call_active` parameter on all SoundStructure devices in the design. When `clink_local_call_active` >= 1, the status LEDs on the Polycom table microphones illuminates green to indicate an active call is in progress
Polycom Video Codec Call Inactive	Decrements the local `clink_local_call_active` parameter on all SoundStructure devices in the design. When `clink_local_call_active` >= 1, the status LEDs on the Polycom table microphones illuminates green to indicate an active call is in progress. When `clink_local_call_active`=0, the status LEDs are turned off.

SoundStructure PSTN Interface Events

The events in the following table are automatically generated when SoundStructure telephony interfaces and Polycom table or ceiling microphones are designed as part of a SoundStructure project.

Automatically Generated Events for SoundStructure Telephony Projects

Event Name	Description
Phone Out Call Connect	Increments the local `clink_local_call_active` parameter on all SoundStructure devices in the design. When `clink_local_call_active` >= 1 on a particular SoundStructure device, the status LEDs on the Polycom table microphones connected to that device illuminates green to indicate an active call is in progress. This event uses the automatically generated Increment Active Call Count preset to increment the number of active calls.
Phone Out Call Disconnect	Decrements the local `clink_local_call_active` parameter on all SoundStructure devices in the design. When `clink_local_call_active` >= 1 on a particular SoundStructure device, the status LEDs on the Polycom table microphones connected to that device illuminates green to indicate an active call is in progress. This event uses the automatically generated Decrement Active Call Count preset to increment the number of active calls.
Clink to Mics Mute	Maps the mute of the Polycom Video Codec to mute microphones in the virtual channel (or group) “Mics” on a SoundStructure device
Table Mic A Mute Event Table Mic B Mute Event Table Mic C Mute Event (an event is created for each Polycom table microphone element)	Maps the mute of a particular Polycom microphone to `clink_mute` which is used to control mute of the Polycom Video Codec. Muting any of the microphones set the state of `clink_mute`. Combining this event with the Clink to Mics Mute event causes the virtual channel (or group) “Mics” to mute when any microphone is muted.

Push To Talk Microphone Events

When adding Push to talk microphones to a project, there are several logic input mode options for what should happen when the button is pressed. There are also several logic output modes available for what the status LED should indicate.

The automatic options for the Logic Input Mode include:

- Toggle microphone mute – toggle the mute on this particular microphone when the button is pressed

- Toggle all microphone mute – toggle the mute on all microphones when the button is pressed
- Push to mute – push and hold this button to mute the microphone (e.g., a cough button)
- Push to talk – push and hold this button to unmute the microphone

The automatic options for the Logic Output Mode include:

- Active on mute – illuminate the LED when the microphone is muted
- Activate on unmute – illuminate the LED when the microphone is unmuted
- Activate on gate – illuminate the LED when the microphone automixer gates on

These options are provided to make it easy to automatically create events based on the desired behavior. These events may be further customized by double clicking on any event to open the Edit Event user control.

Depending on the selected logic input and output behavior, different events are created. The following table summarizes the events created for the different logic input mode options.

Logic Input Mode Events

Logic **Input** Mode	Event Name	Description
Toggle Mic Mute	Table Mic Button Event	When the microphone switch is closed, toggles the mute on this particular microphone. Does nothing when the switch is opened.
Toggle All Mics Mute	Table Mic Mute Event	This event maps the mute state of the microphone to `clink_mute`. If the microphone is muted (mute=1), then the value of `clink_mute` is set to 1 on SoundStructure device 1. If the microphone is unmuted (mute=0) then the value of `clink_mute` is set to 0 on SoundStructure device 1. There is another event, Clink to Mics Mute which maps the `clink_mute` state on this device to mute the virtual channel (or group) “Mics”. The net result is that the mute state of all microphones in the group “Mics” are toggled when the switch is closed.
	Table Mic Button Event	When the microphone switch is closed, toggles the value of `clink_mute`.
	Clink to Mics Mute	Maps the mute of the Polycom Video Codec to mute “Mics” on SoundStructure when all microphones are to be muted
	Clink Mute Link 1 to 2 Clink Mute Link 2 to 1 …	If there are multiple SoundStructure devices in a design, there are events to synchronize the clink_mute state of each device to the next device and from the last device to the first device.

Logic Input Mode Events

Push-to-mute	Table Mic Button Event	This event maps the value of the `digital_gpio_state` of the button inversely to the mute state of the microphone. While the microphone switch is closed (`digital_gpio_state`=0), the microphone is muted (mute=1). While the microphone switch is open (`digital_gpio_state`=1), the microphone is muted (mute=0).
Push-to-talk	Table Mic Button Event	This event maps the value of the `digital_gpio_state` of the button directly to the mute state of the microphone. While the microphone switch is closed (`digital_gpio_state`=0), the microphone is unmuted (mute=0). While the microphone switch is open (`digital_gpio_state`=1), the microphone is muted (mute=1).

Depending on the logic output mode, there are additional events that are generated as summarized in the table below.

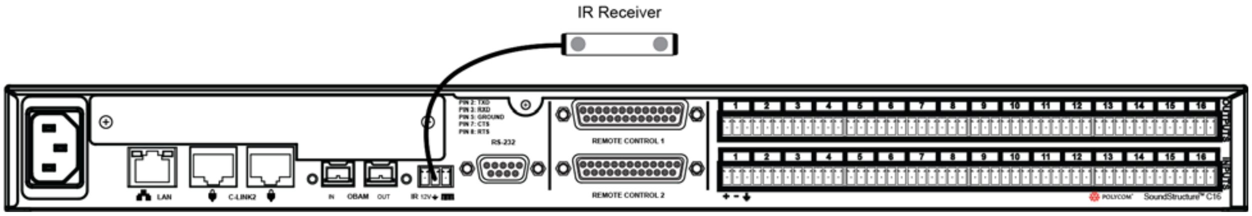
Generated Logic Output Mode Events

Logic **Output** Mode	Event Name	Description
Activate on Mute	Table Mic LED Event	This event maps the mute state of the microphone directly to the digital_gpio_state of the LED. If the microphone is muted (mute=1) then the LED is turned on (digital_gpio_state=1).
Activate on Unmute	Table Mic LED Event	This event maps the mute state of the microphone inversely to the digital_gpio_state of the LED. If the microphone is muted (mute=1) then the LED is turned off (digital_gpio_state=0. If the microphone is unmuted (mute=0) then the LED is turned on (digital_gpio_state=1).
Activate on Gate	Table Mic LED Event	This event maps the gate state (am_gate parameter) directly to the digital_gpio_state of the LED. If the microphone gates on (am_gate=1), the LED is turned on (digital_gpio_state=1). If the microphone gates off (am_gate=0), the LED is turned off (digital_gpio_state=0).

Polycom IR Remote

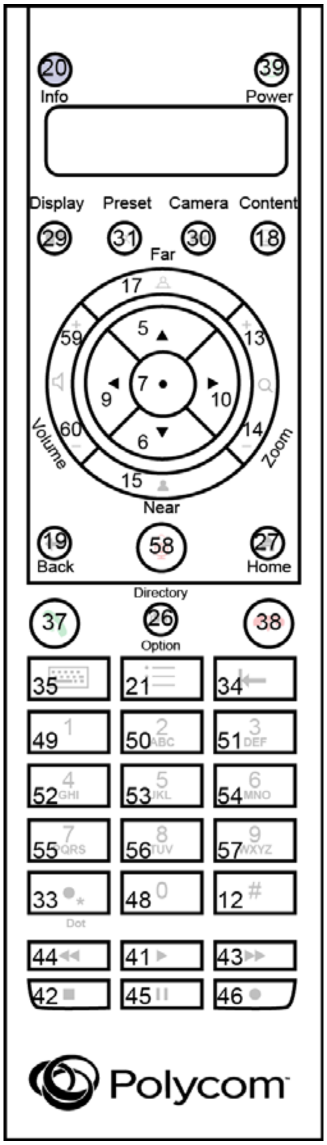
In stand-alone SoundStructure applications without an Polycom Video Codec, the Polycom IR remote can be used to control SoundStructure devices that have both version 1.3 firmware and an external IR remote receiver. To use the IR remote transmitter, add a Polycom IR remote to the project and connect the receiver physically to the SoundStructure as shown in the following figure.

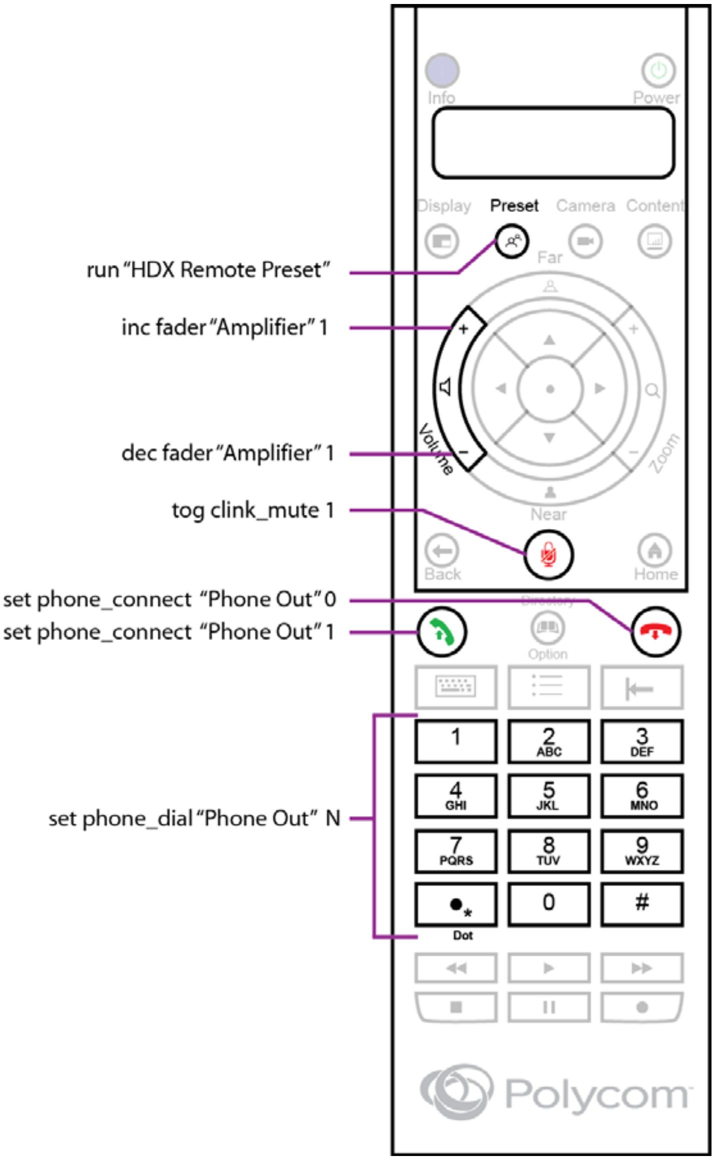
IR Remote Receiver Connected to SoundStructure Device


IR Receiver
PIN 2: TXD
PIN 3: RXD
PIN 5: GROUND
PIN 7: CTS
PIN 8: RTS
RS-232
REMOTE CONTROL 1
REMOTE CONTROL 2
LAN
C-LINK2
IN
OBAM
OUT
IR 12V
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
OUTPUTS
INPUTS
+ −
POLYCOM
SoundStructure™ C16

All the keys on the IR remote may be used as sources of events. The individual keys are selected in an event by specifying a trigger that is equal to the key of interest. The entire set of key presses that may be defined are shown in the following figure (left) and the default key mappings that are created automatically are shown on the right.

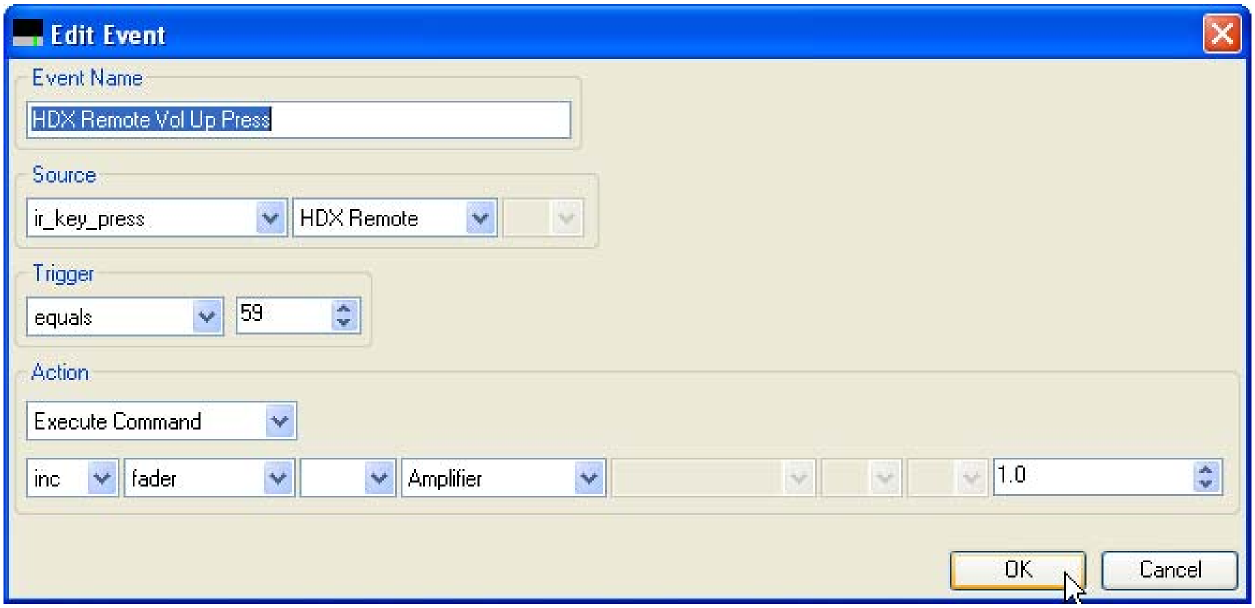
Defined Key Presses and Default Key Mapping


20
Info
39
Power
Display
Preset
Camera
Content
29
31
30
18
Far
17
5
59
13
9
7
10
60
6
14
15
Volume
Zoom
Near
19
Back
58
27
Home
Directory
37
26
Option
38
35
21
34
49
1
50
2 ABC
51
3 DEF
52
4 GHI
53
5 JKL
54
6 MNO
55
7 PQRS
56
8 TUV
57
9 WXYZ
33
Dot
48
0
12
#
44
41
43
42
45
46
Polycom


run "HDX Remote Preset"
inc fader "Amplifier" 1
dec fader "Amplifier" 1
tog clink_mute 1
set phone_connect "Phone Out" 0
set phone_connect "Phone Out" 1
set phone_dial "Phone Out" N
Info
Power
Display
Preset
Camera
Content
Far
Volume
Zoom
Near
Back
Home
Directory
Option
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
0
#
Dot
Polycom

As an example, consider the event for the adjusting the volume of the system. In this example, the trigger equals 59 which is the value of the volume up key on the remote. When key 59 is pressed, the fader for the amplifier is incremented by 1dB.

Adjusting Events for System Volume



Event Name: HDX Remote Vol Up Press

Source: ir_key_press, HDX Remote

Trigger: equals 59

Action: Execute Command

inc, fader, Amplifier, 1.0

The full list of events created when and Polycom IR Remote is added to a project is shown in the following table.

Events Created for Polycom IR Remote

Event Name	Description
IR Remote Vol Down Held	Decrements the fader for “Amplifier” by 1dB
IR Remote Vol Down Press	Decrements the fader for “Amplifier” by 1dB
IR Remote Vol Up Held	Increments the fader for “Amplifier” by 1dB
IR Remote Vol Up Press	Increments the fader for “Amplifier” by 1dB
IR Remote Phone 0	Dials the digit “0”
IR Remote Phone 1	Dials the digit “1”
IR Remote Phone 2	Dials the digit “2”
IR Remote Phone 3	Dials the digit “3”
IR Remote Phone 4	Dials the digit “4”
IR Remote Phone 5	Dials the digit “5”
IR Remote Phone 6	Dials the digit “6”
IR Remote Phone 7	Dials the digit “7”
IR Remote Phone 8	Dials the digit “8”
IR Remote Phone 9	Dials the digit “9”

Events Created for Polycom IR Remote

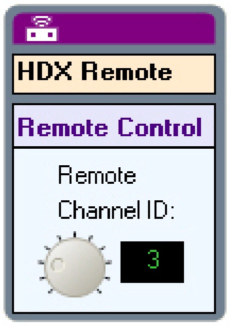
IR Remote Phone *	Dials the digit “*”
IR Remote Phone #	Dials the digit “#”
IR Remote Phone Disconnect	Hangs up the phone
IR Remote Phone Connect	Takes the phone offhook
IR Remote Mute	Toggles the state of clink_mute which is used to mute all microphones
IR Remote Preset Press	Runs the preset “Polycom IR Remote Preset”

The automatically generated events may be customized to suit a particular application and additional events for the other key presses on the Polycom IR remote may be added by using the Add Events feature.

Polycom IR Remote Channel ID

SoundStructure Studio creates projects assuming the Polycom IR remote has the default Channel ID of 3. Changing the default value from 3 to an alternative value may be done on the logic page by adjusting the knob of the channel ID from 0 to 15.

Polycom IR Remote Channel ID

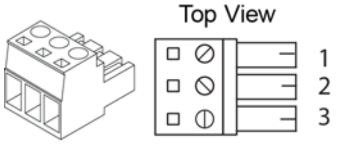


IR Receiver Connector

To use a Polycom IR remote transmitter, the SoundStructure system requires an IR receiver. Each SoundStructure device includes an IR receiver interface port that can be used with IR receivers from Xantech including models 780-80, 780-90, 480-00, 480-80, and 490-00. The IR receiver should be connected to the SoundStructure device using the pin out shown in the following figure.

The SoundStructure device supplies 12V so the receiver can be connected directly to the IR port on the SoundStructure device without an external power supply.

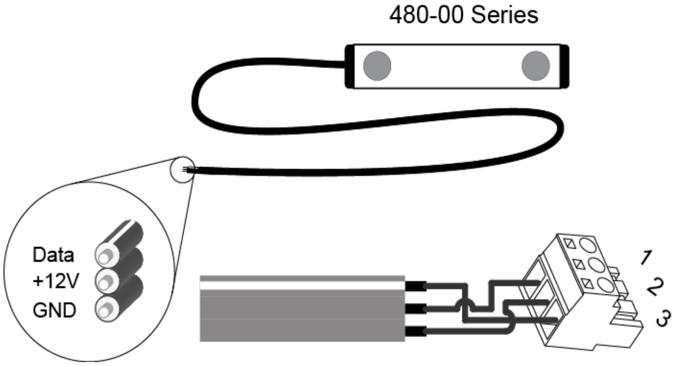
IR Receiver Connected to SoundStructure Device with Pin Out



IR Receiver	
Pin	**Signal**
1	+12V
2	Ground
3	IR Signal Data

The wiring for the typical Xantech receiver is shown in the following figure.

Xantech Receiver Wiring



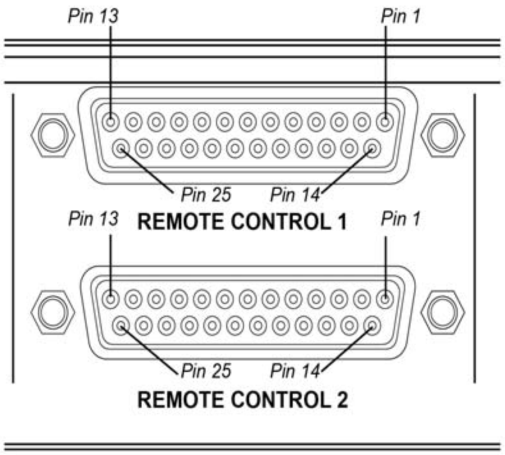
Logic Ports

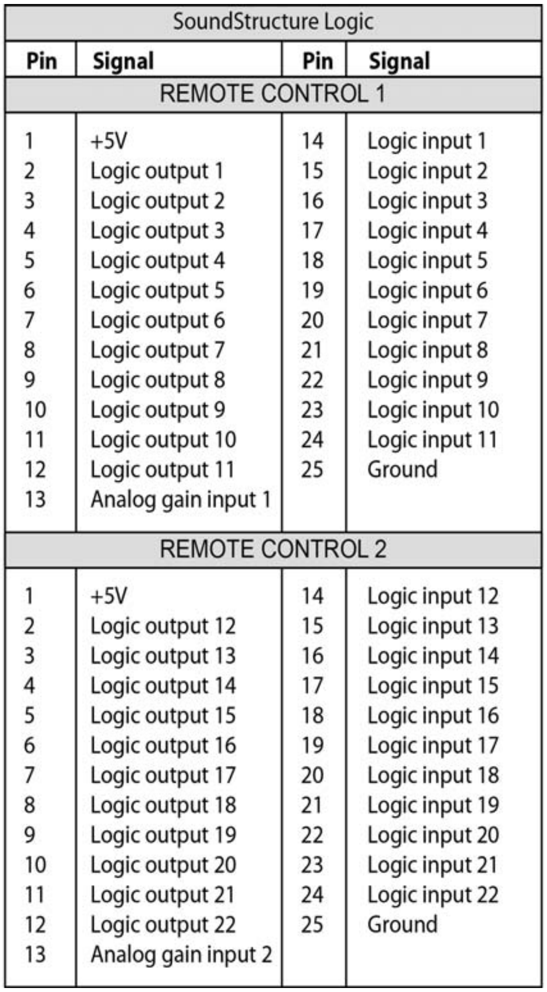
Each SoundStructure device has two DB25 connectors where each DB25 connector has

- 11 logic inputs on each connector for a total of 22 logic inputs.
- 11 logic outputs on each connector for a total of 22 logic outputs
- One analog logic input on each connector for a total of two analog logic inputs
- One 5V supply on each connector for a total of two 5V supply pins
- One signal ground on each connector for a total of two signal grounds

The pin out of the rear-panel DB25 connectors is shown in the figure.

Rear Panel DB25 Connector Pin Out





SoundStructure Logic

Pin	Signal	Pin	Signal
REMOTE CONTROL 1			
1	+5V	14	Logic input 1
2	Logic output 1	15	Logic input 2
3	Logic output 2	16	Logic input 3
4	Logic output 3	17	Logic input 4
5	Logic output 4	18	Logic input 5
6	Logic output 5	19	Logic input 6
7	Logic output 6	20	Logic input 7
8	Logic output 7	21	Logic input 8
9	Logic output 8	22	Logic input 9
10	Logic output 9	23	Logic input 10
11	Logic output 10	24	Logic input 11
12	Logic output 11	25	Ground
13	Analog gain input 1		
REMOTE CONTROL 2			
1	+5V	14	Logic input 12
2	Logic output 12	15	Logic input 13
3	Logic output 13	16	Logic input 14
4	Logic output 14	17	Logic input 15
5	Logic output 15	18	Logic input 16
6	Logic output 16	19	Logic input 17
7	Logic output 17	20	Logic input 18
8	Logic output 18	21	Logic input 19
9	Logic output 19	22	Logic input 20
10	Logic output 20	23	Logic input 21
11	Logic output 21	24	Logic input 22
12	Logic output 22	25	Ground
13	Analog gain input 2		

Logic inputs have a value that is read as either 0 or 1, logic outputs have a value that is set to either 0 or 1, and the analog gain inputs have a value that varies from 0 to 255. The details of how to use logic pins are described in the following sections.

Digital Logic Inputs

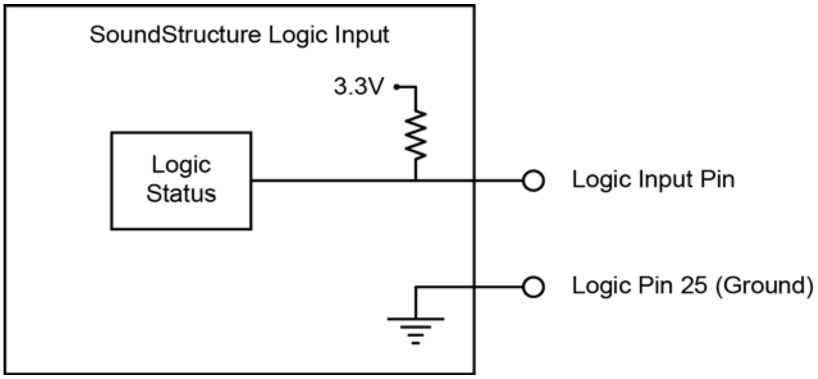
Logic inputs allow one to connect push to talk buttons and other dry contact[1] closures to the rear-panel of a SoundStructure device. The circuitry behind each logic input, shown in the following figure, shows that the logic inputs have a default value of 1 due to the internal pull-up resistor.

---

[1] A dry contact closure is one where there is no voltage externally applied to the contacts – it is simply an open or closed circuit.

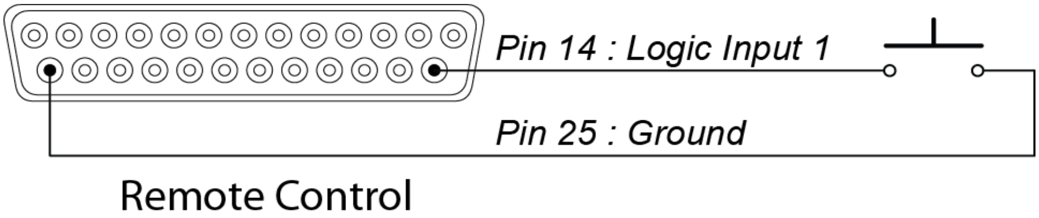
The logic inputs has a default value of 1 (high) when the contact closure is open, and has a value of 0 (low) when the contact closure is closed and tied to ground.

Logic Input Circuitry


SoundStructure Logic Input
3.3V
Logic Status
Logic Input Pin
Logic Pin 25 (Ground)

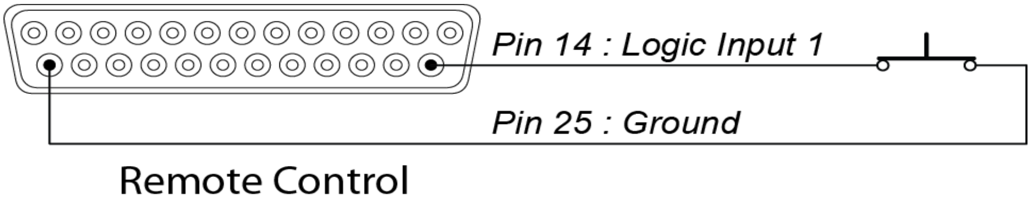
A typical contact closure example is shown in the following two figures. In the first figure, the input generates the value 1 (high) because the switch is open.

Open Remote Control Contact Closure Example


Pin 14 : Logic Input 1
Pin 25 : Ground
Remote Control

When the logic switch is closed, as shown in the figure below, the logic value reads the value 0 (low) indicating that the contact has been closed.

Closed Remote Control Contact Closure Example


Pin 14 : Logic Input 1
Pin 25 : Ground
Remote Control

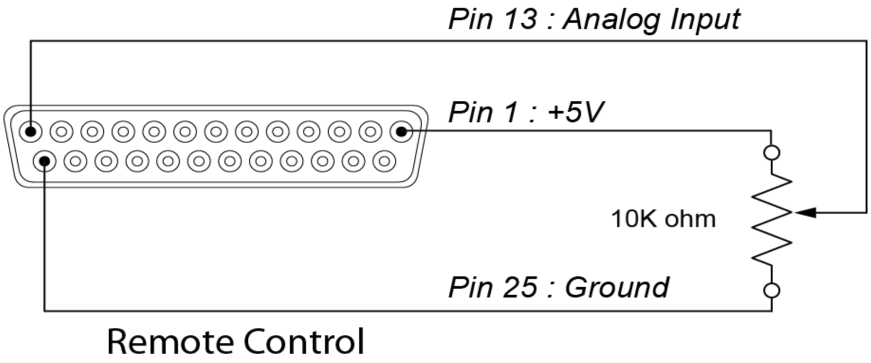
The logic inputs are internally debounced and can detect changes in the contact closures as short as 100msec.

Analog Logic Inputs

The analog gain inputs (analog gain 1 and 2) operate by measuring an analog voltage between the analog input pin and the ground pin. The maximum input voltage level should not exceed +6 V. It is recommended that the +5 V supply on Pin 1 be used as the upper voltage limit.

An example of connecting a volume knob potentiometer for volume control is shown in the following figure. In this figure, the volume knob has three connections – one to the +5V connection, one to ground, and the third, the wiper of the potentiometer, will be connected to the analog gain input. As the knob is turned, the voltage measured varies between 0 and approximately 5V. The values measured from the analog logic gain input varies from approximately 0 to 255.

Connected Volume Knob Potentiometer for Volume Control

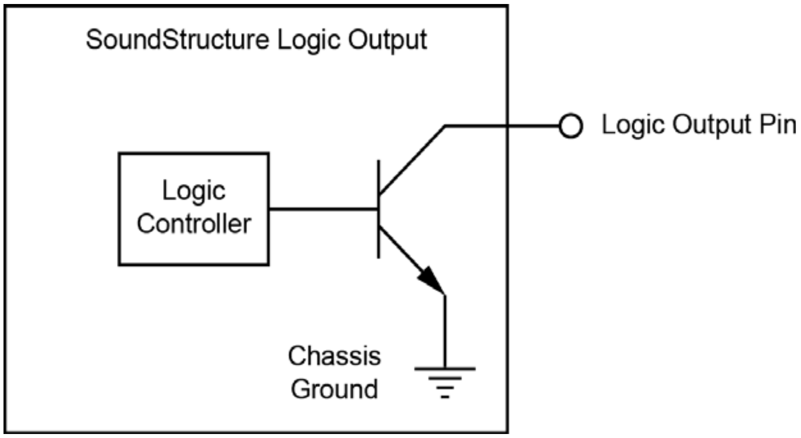


Logic Outputs

SoundStructure devices implement logic outputs as open collector circuits. The open collector design, shown in the following figure, makes it possible to drive LED’s and relays with minimal additional circuitry. Please note that only positive external voltages, such as the +5V supply on pin 1, should be used with the logic output pin. Each logic output pin is capable of sinking 60mA of current.

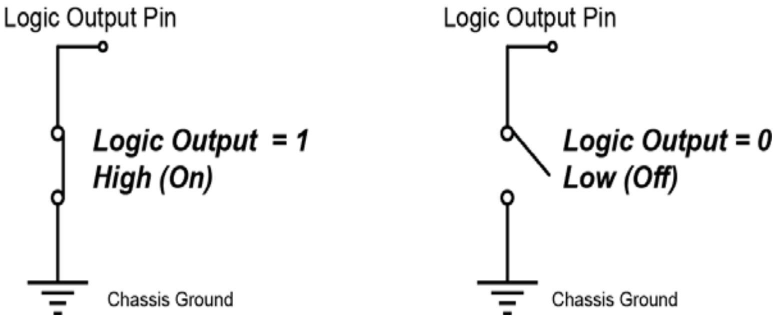
If using an external voltage supply as part of any logic output circuit, the maximum voltage that should be used with the logic outputs is 60V with a maximum current of 500 mA.

Open Collector Logic Output Design


SoundStructure Logic Output
Logic Output Pin
Logic Controller
Chassis Ground

As shown in the following figure, when the logic output pin is set to 1 the output pin allows current to flow from the logic output pin to the chassis ground, thus completing a signal circuit path. When the logic output is set to 0, no current flows from the logic output pin to ground and the circuit is open.

Flow from Logic Output Pin to Chassis Ground


Logic Output Pin
Logic Output = 1
High (On)
Chassis Ground
Logic Output Pin
Logic Output = 0
Low (Off)
Chassis Ground

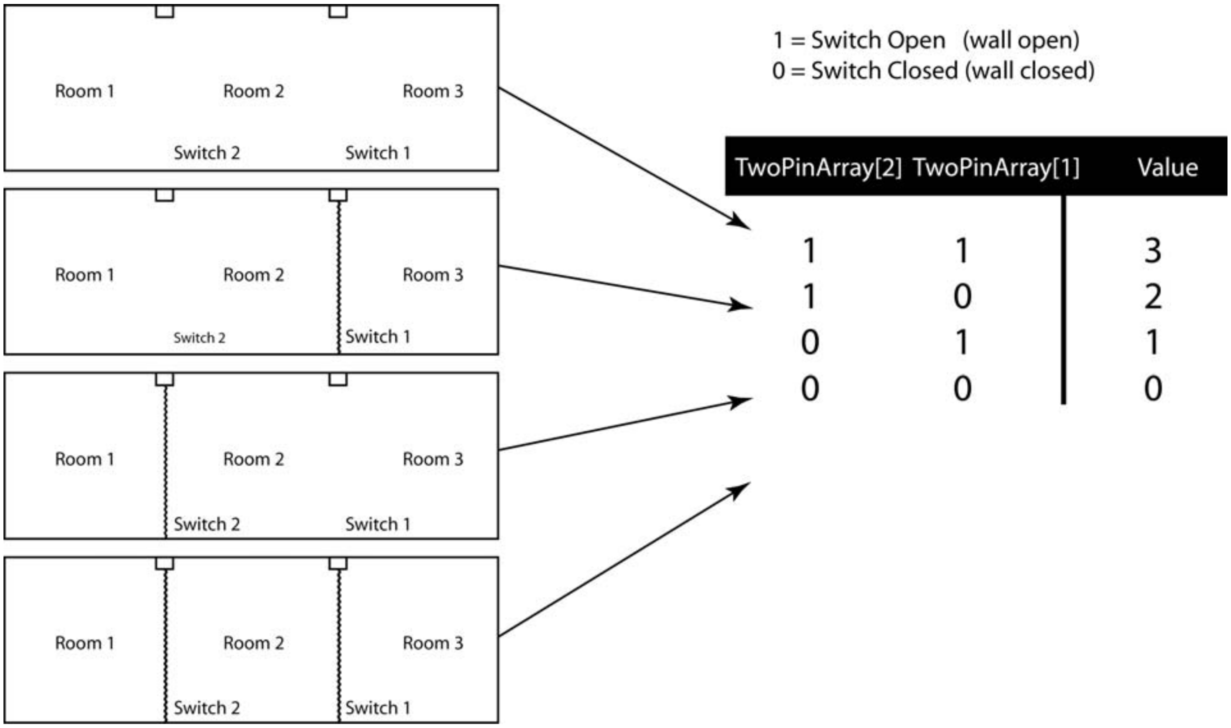
Logic Arrays

It is possible to link multiple logic pins together in a logic array. A logic input array is useful when there are more than two logic states that are important. For example, in a split and combine room with two movable partitions, there are four different combinations that must be considered as shown in the following figure with a logic input array that consists of two input pins. These two pins allow all four combinations of the room partitions to be specified. In the Understanding Events section, we'll see how to use the logic array values as sources and execute different presets based on the value of the logic array.

When defining logic array pins, the pin with the highest array index is the most significant bit. As shown in the figure, creating a two pin logic input array creates TwoPinArray[2] and TwoPinArray[1] pins as part of the array TwoPinArray. TwoPinArray[2] is the most significant bit in the two bit word.

The value of a logic input array is read with the `digital_gpio_value` parameter. If a logic pin is part of an array, it may not also be used as an individual logic input pin.

Defining Logic Arrays for TwoPinArray



1 = Switch Open (wall open)
0 = Switch Closed (wall closed)

TwoPinArray[2]	TwoPinArray[1]	Value
1	1	3
1	0	2
0	1	1
0	0	0

Logic output arrays may also be defined. The value of a logic output array is set using the digital_gpio_value parameter. For example if three logic output pins are part of a logic output array, the command

set digital_gpio_value “Output Array” 7

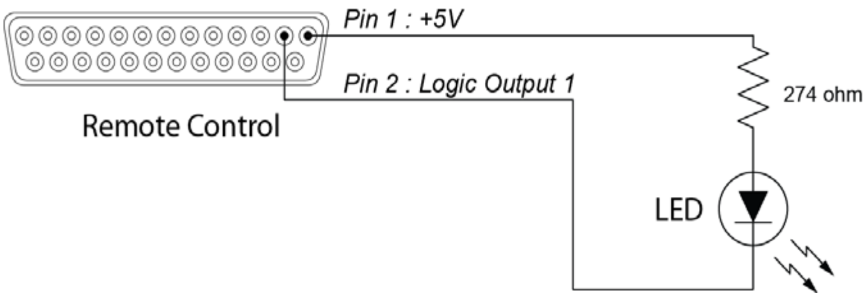
sets all of the pins in the array named “Output Array” to the value 1.

Viewing the LED Example

The following figure is an example of how to use an external LED. Most standard LEDs require approximately 2.0 V to illuminate. In this example a 274 ohm resistor is used to limit the current from the 5V supply of Pin 1and to limit the voltage and current to a safe level for the LED. Increasing the series resistor value decreases the current through the circuit and also decreases the voltage at the input to the LED, reducing the brightness of the LED.

When the logic output is set to 1, current flows and the LED turns on. When the logic output is set to 0, current does not flow and the LED turns off.

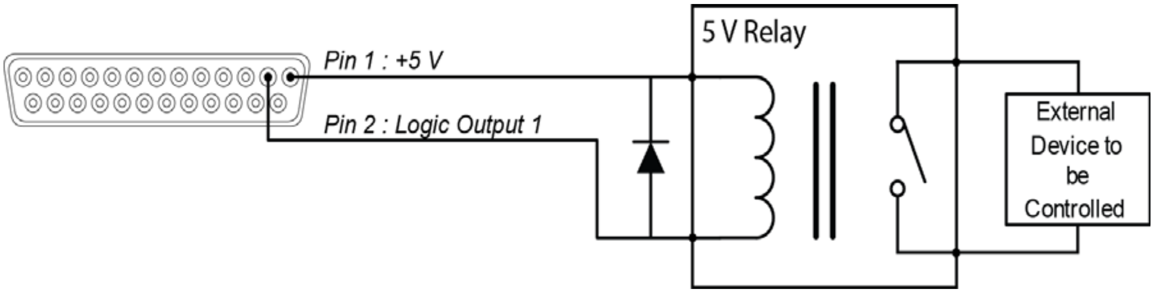
External LED Remote Control



Viewing the Relay Example

The following figure is an example of how to drive a 5V relay. When the logic output (Pin 2 in this example) is set to 1, current flows from Pin 2 to ground and that current flow energizes the relay coil and close the relay contact. When the logic output is set 0, current stops flowing to the relay coil, causing the relay contact to open. A diode is recommended to be placed in parallel with the relay to provide a path for the discharge current of the magnetic coil of the relay. This current discharges over a very short period of time and a diode capable of handling a large amount of surge current such as the 1N4001 is recommended and is available from several manufacturers. This example circuit uses an Omron G5CA relay and the coil resistance is 125 ohms. Because of this coil resistance, an additional series resistor is not required to limit the current from the 5 V supply to less than 500 mA in this example.

Driving a 5V Relay



Viewing Event Examples

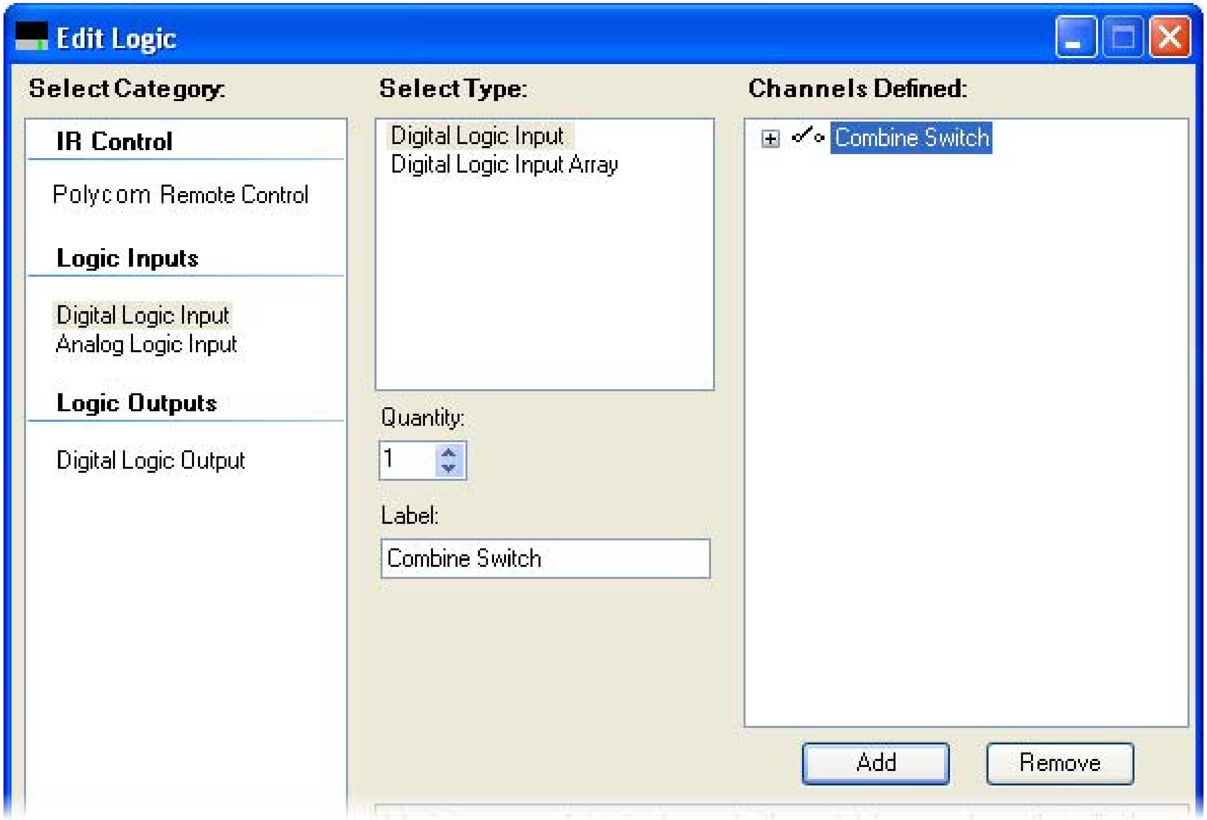
This section provides several examples of how to use events to customize a SoundStructure design.

Splitting and Combining Presets Triggered from a Logic Input

In this example, two presets are selected in the SoundStructure device, “Combine” and “Split”. The goal of this application is to have a logic input select which preset is executed based on the state of the logic switch.

Step 1: Use the Edit Logic button to add a single Digital Logic Input.

From the Events page select the **Edit logic** button and select a single digital logic input pin. Use a name for the pin that makes sense to you. In this example the pin is called “Combine Switch”.


Edit Logic

Select Category:

IR Control
- Polycom Remote Control

Logic Inputs
- Digital Logic Input
- Analog Logic Input

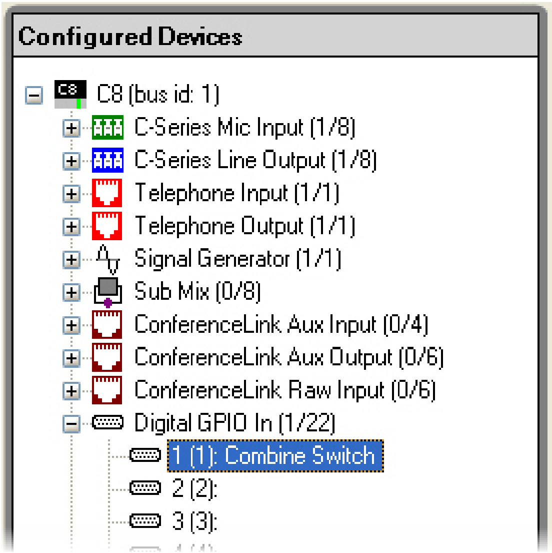
Logic Outputs
- Digital Logic Output

Select Type:
- Digital Logic Input
- Digital Logic Input Array

Quantity: 1

Label: Combine Switch

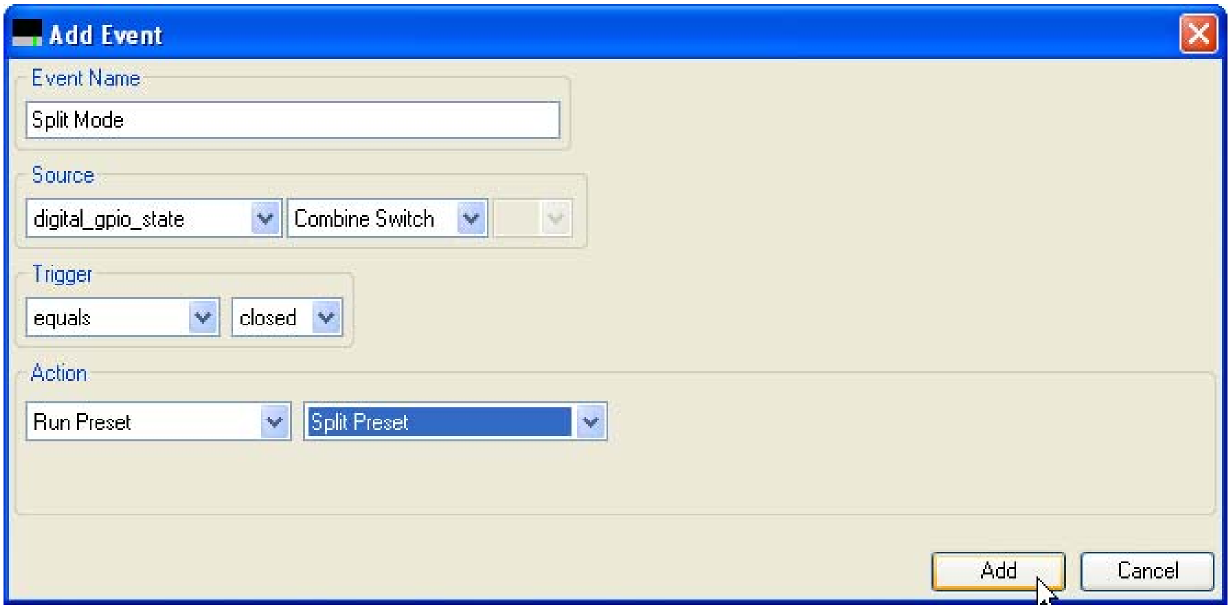
Channels Defined:
- Combine Switch


Configured Devices

- C8 (bus id: 1)
  - C-Series Mic Input (1/8)
  - C-Series Line Output (1/8)
  - Telephone Input (1/1)
  - Telephone Output (1/1)
  - Signal Generator (1/1)
  - Sub Mix (0/8)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (1/22)
    - 1 (1): Combine Switch
    - 2 (2):
    - 3 (3):

Step 2: Create events for both the split and combined mode

The next step is to determine how the logic switch operates – when the switch circuit is closed are the rooms split or combined? This information determines which preset is called when the switch is open and when the switch is closed. Assuming that the split preset is required when the switch is closed, the following Split Mode and Combine Mode events should be created.

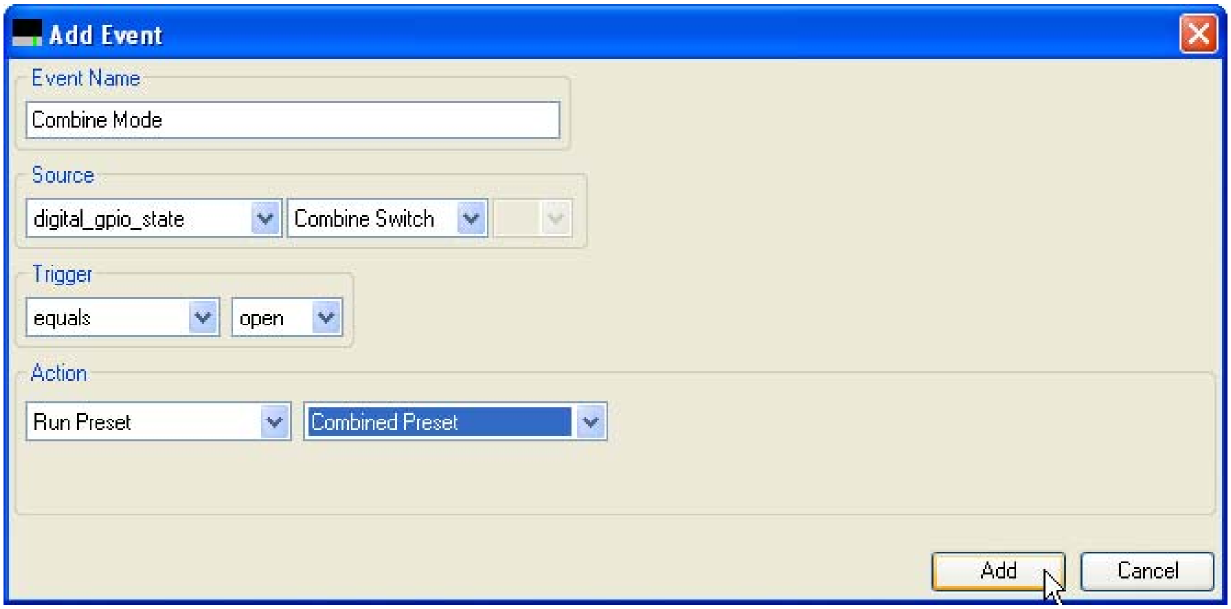


Event Name: Split Mode

Source: digital_gpio_state, Combine Switch

Trigger: equals, closed

Action: Run Preset, Split Preset



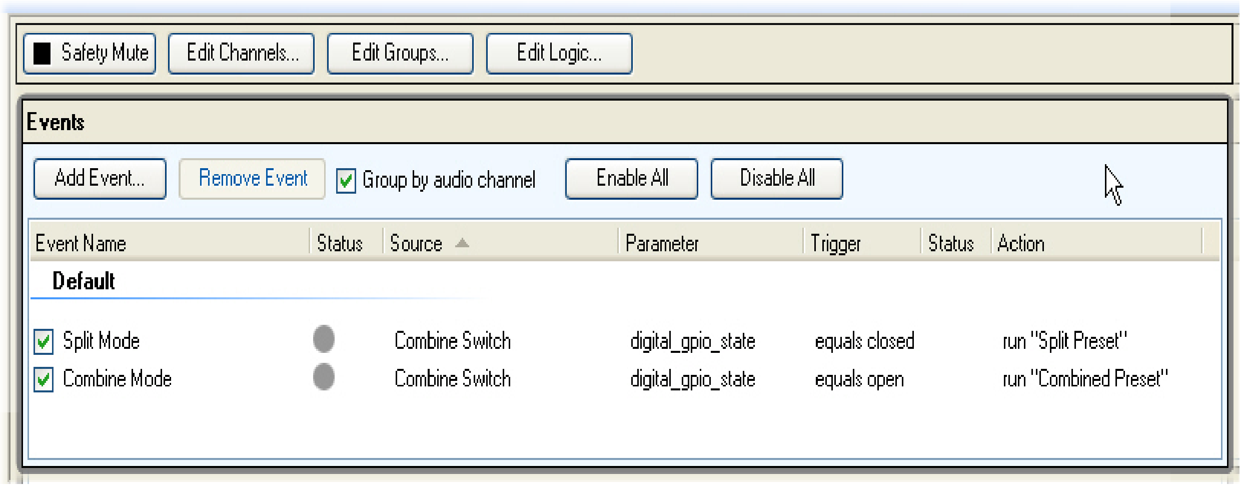
Event Name: Combine Mode

Source: digital_gpio_state, Combine Switch

Trigger: equals, open

Action: Run Preset, Combined Preset

After adding these two events, the event page shows both events. From the events page is it easy to see that when the switch is closed the split preset are run and when the switch is open the combined preset is run. If it is determined once the switch is installed that the switch logic is reversed and the split preset should be called when the switch is open and the combined preset when the switch is closed, then the triggers or the actions may be easily reversed on the events. To edit an event, double click the event.


■ Safety Mute
Edit Channels...
Edit Groups...
Edit Logic...
Events
Add Event...
Remove Event
☑ Group by audio channel
Enable All
Disable All

Event Name	Status	Source	Parameter	Trigger	Status	Action
Default						
☑ Split Mode	●	Combine Switch	digital_gpio_state	equals closed		run "Split Preset"
☑ Combine Mode	●	Combine Switch	digital_gpio_state	equals open		run "Combined Preset"

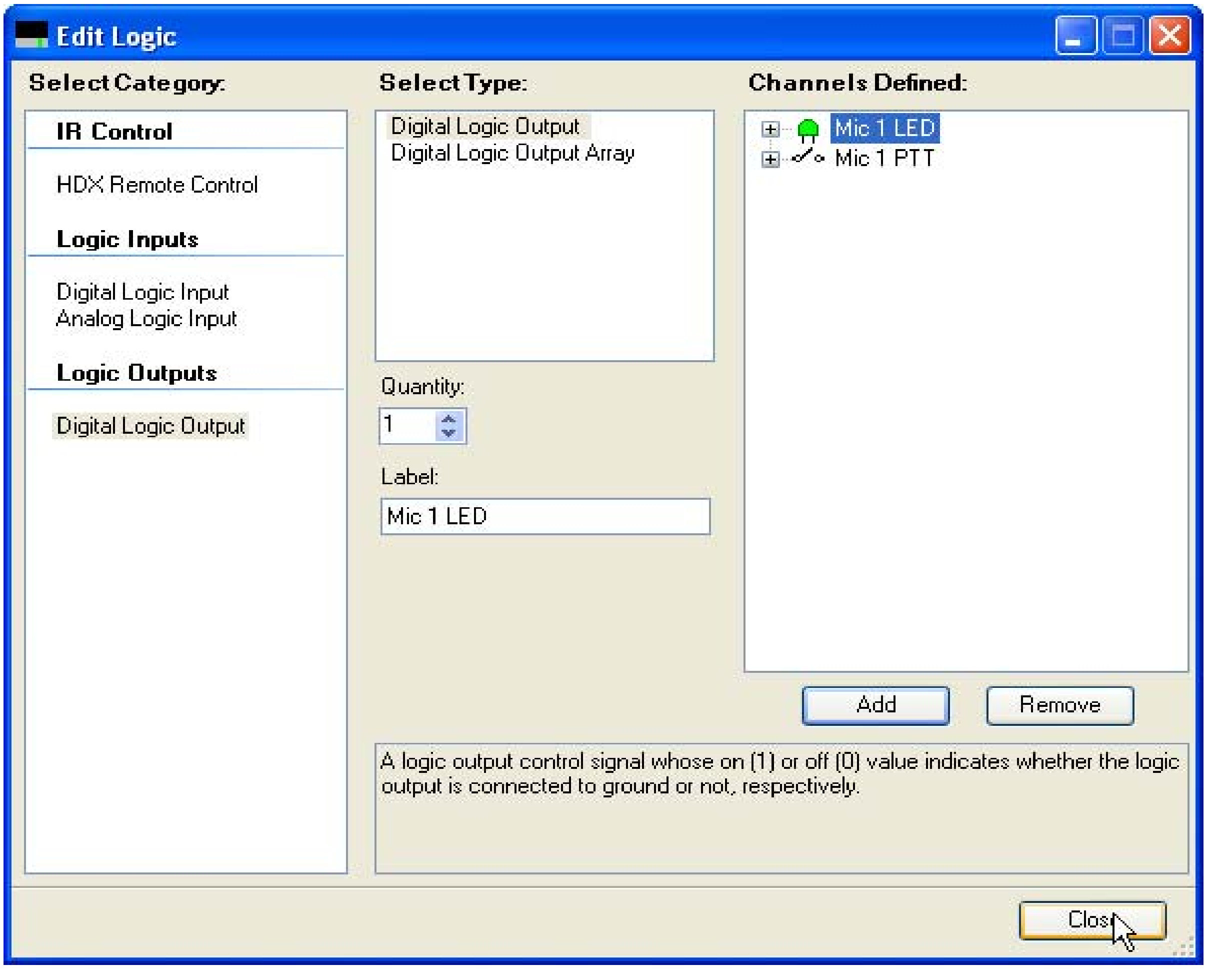
Viewing Push To Talk Microphones with LEDs Example

In this example, the push to talk button on a microphone is used to mute all microphones in room and the status LED on the microphone is illuminated when the microphone is unmuted.

This example assumes that the microphone is already part of the system and now the logic inputs and outputs are manually added to the existing system. If this is a new system, then use the logic input and logic output modes on the edit channels control to automatically add the logic inputs and outputs and events when the microphone inputs are defined.

Step 1: Add a logic input and output pin for each microphone.

Use names for the logic pins that makes sense to you as you build your system. If you have many microphones, you may add multiple digital logic inputs and outputs by adjusting the quantity before clicking the add button.


Edit Logic
Select Category:
IR Control
HDX Remote Control
Logic Inputs
Digital Logic Input
Analog Logic Input
Logic Outputs
Digital Logic Output
Select Type:
Digital Logic Output
Digital Logic Output Array
Channels Defined:
Mic 1 LED
Mic 1 PTT
Add
Remove
Close

Quantity: 1

Label: Mic 1 LED

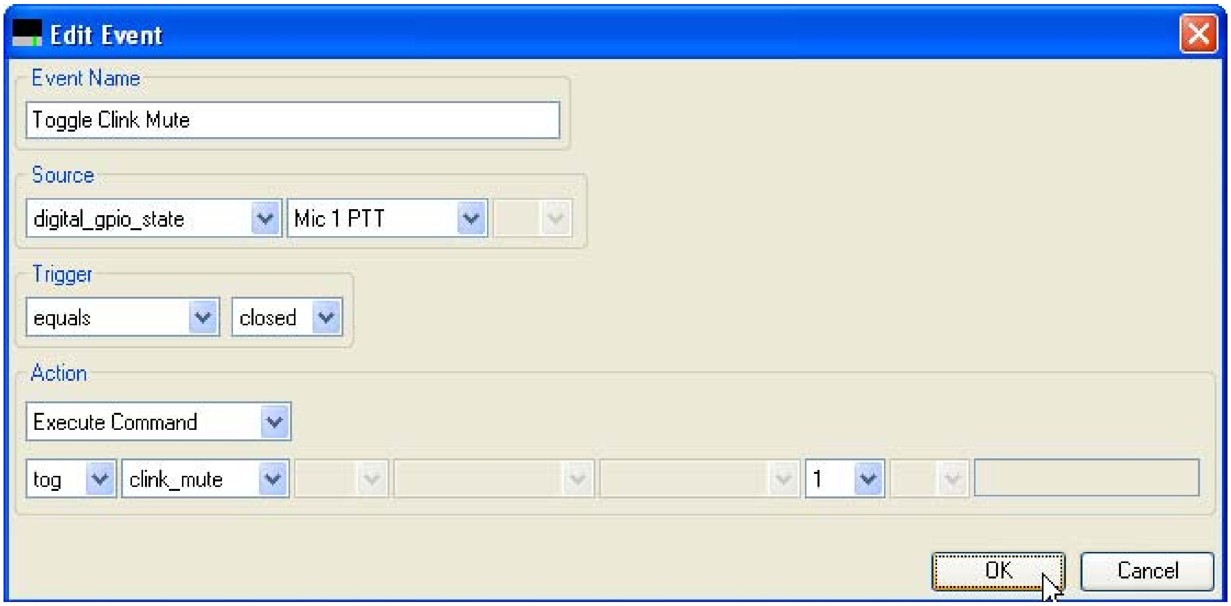
A logic output control signal whose on (1) or off (0) value indicates whether the logic output is connected to ground or not, respectively.

Step 2: Create Mute Events on the button push

In this example it was desired to have the mute state toggled on all microphones when the PTT button is pushed. To accomplish this, create two events – one to toggle the `clink_mute` parameter and one to use the clink_mute parameter to mute “Mics”.

The first event “Toggle Clink Mute” toggles the state of the `clink_mute` parameter on SoundStructure device 1. If there are multiple SoundStructure devices in the system, then additional events are created to map `clink_mute` on device 1 to clink_mute on device 2 and so on and finally map the `clink_mute` on device N (N may vary from 2 to 8) back to the `clink_mute` on device 1 to ensure the mute state is synced across multiple SoundStructure devices.

In this event, every time the PTT button is closed, the clink_mute parameter is toggled.



Event Name: Toggle Clink Mute

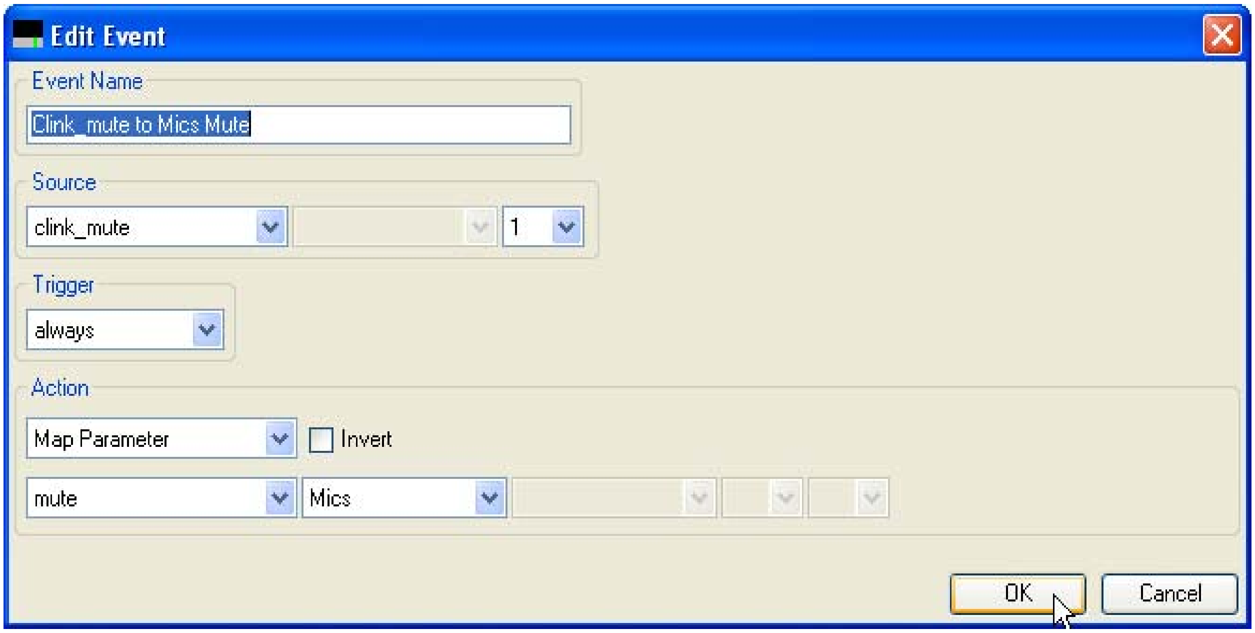
Source: digital_gpio_state, Mic 1 PTT

Trigger: equals, closed

Action: Execute Command

tog, clink_mute, 1

To mute the microphones based on `clink_mute`, another event is required to map `clink_mute` to the mute of the “Mics” group.



Event Name: Clink_mute to Mics Mute

Source: clink_mute, 1

Trigger: always

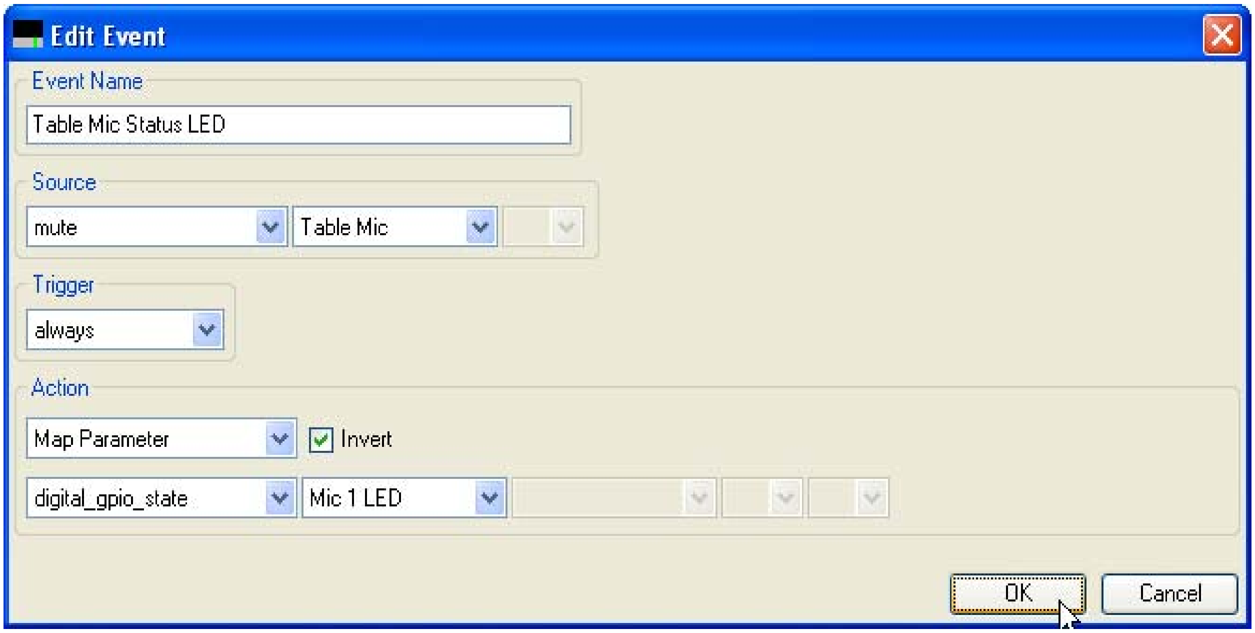
Action: Map Parameter ☐ Invert

mute, Mics

The Clink_mute to Mics Mute event take the clink_mute state of device 1 and maps that value to the mute state of the Mics group. Now whenever the PTT microphone button is pushed, the clink_mute parameter toggles and any change in the clink_mute parameter is mapped directly to the mute state of “Mics” causing all microphones in the “Mics” group to be muted.

Step 3: Create LED event based on the Mute state

The final event required maps the mute state of the microphone to the LED state. In this example, it is desired to light the LED (logic output = 1) when the microphone is unmuted (mute = 0). To accomplish this, use an invert in the map action, as shown in the following figure.



Event Name: Table Mic Status LED

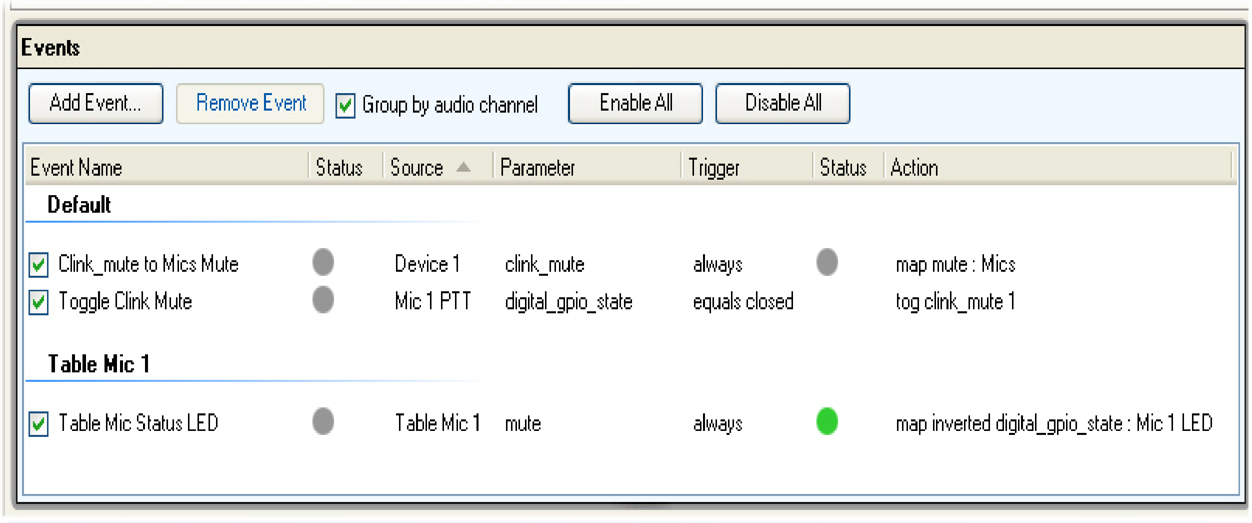
Source: mute | Table Mic

Trigger: always

Action: Map Parameter ☑ Invert

digital_gpio_state | Mic 1 LED

The full set of events is shown on the Events page and displays as the figure shown below.



Event Name	Status	Source	Parameter	Trigger	Status	Action
Default						
☑ Clink_mute to Mics Mute	●	Device 1	clink_mute	always	●	map mute : Mics
☑ Toggle Clink Mute	●	Mic 1 PTT	digital_gpio_state	equals closed		tog clink_mute 1
Table Mic 1						
☑ Table Mic Status LED	●	Table Mic 1	mute	always	🟢	map inverted digital_gpio_state : Mic 1 LED

For each additional microphone that it is desired to add PTT logic, create additional Toggle Clink Mute and Table Mic Status LED events for each microphone. Only one “Clink_mute to Mics Mute” event is required.

Remember, if you have microphones across multiple SoundStructure devices you will require events that will map the clink_mute of device 1 to the clink_mute of device 2, and so on to ensure the clink_mute parameter on all devices is synchronized together.

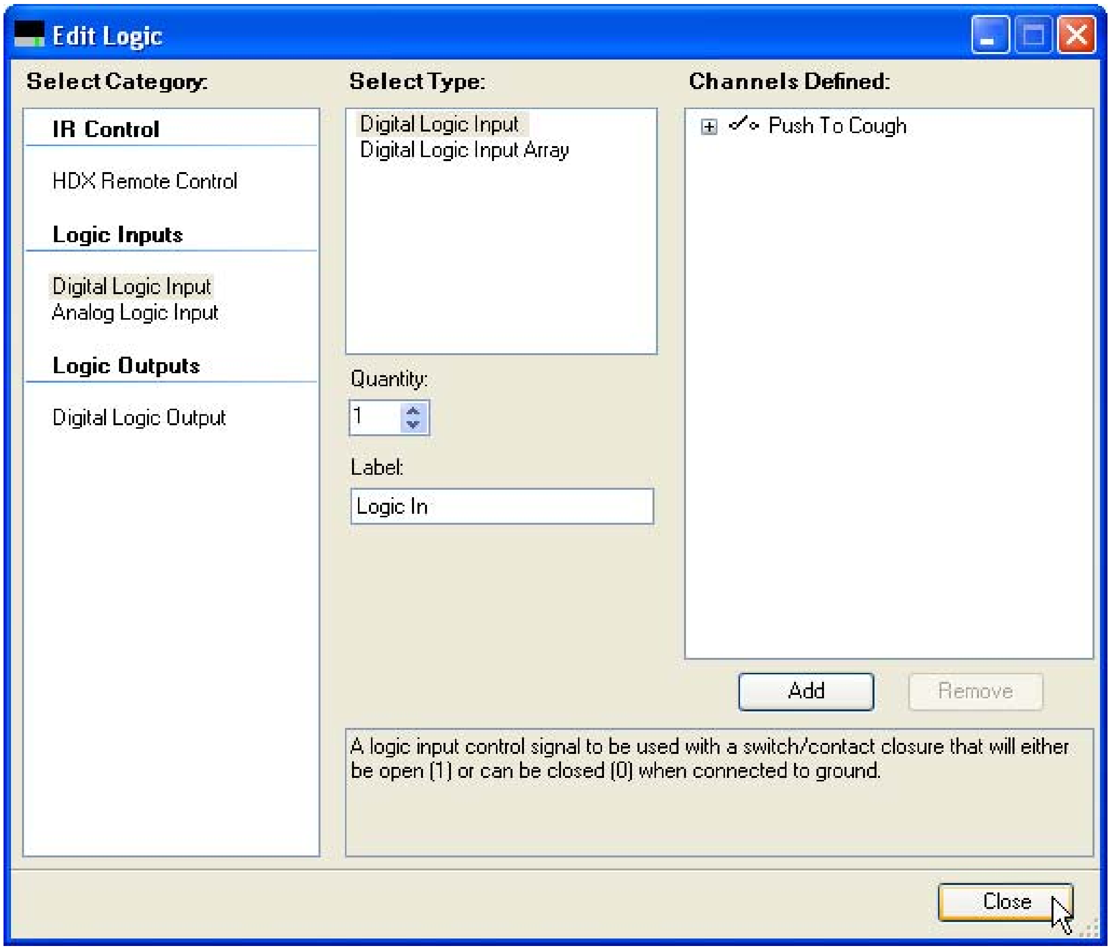
Viewing Push and Hold to Temporarily Mute A Microphone

In this example, a cough button is created to allow someone to mute their particular microphone while they are holding the button closed.

This example assumes that the microphone is already part of the system and now the logic inputs and outputs will be manually added to the existing system. If this is a new system, then use the logic input and logic output modes on the edit channels control to automatically add the logic inputs and outputs and associated events when the microphone input is defined.

Step 1: Add the logic input button.

Use the edit logic button to add a digital logic input.


Edit Logic
Select Category:
IR Control
HDX Remote Control
Logic Inputs
Digital Logic Input
Analog Logic Input
Logic Outputs
Digital Logic Output
Select Type:
Digital Logic Input
Digital Logic Input Array
Channels Defined:
Push To Cough
Add
Remove
Close

Quantity: 1

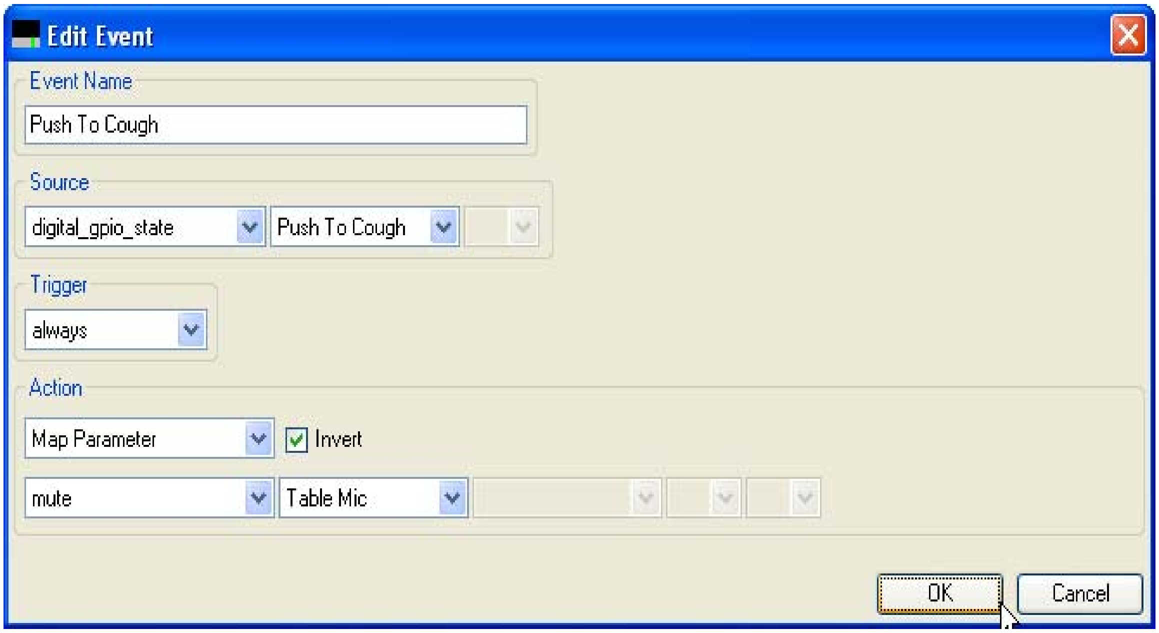
Label: Logic In

A logic input control signal to be used with a switch/contact closure that will either be open (1) or can be closed (0) when connected to ground.

As with other examples, check the wiring page to confirm that the logic input is on the desired logic input pin.

Step 2: Create the event to map the button press to the mute state of the microphone

Since the microphone should be muted (mute = 1) when the button is pressed (logic input = 0), the event should use an action map with the invert option as shown in the following figure. Anytime the button is pressed, the microphone will be muted. When the button is released, the microphone will be unmuted.



Event Name: Push To Cough

Source: digital_gpio_state | Push To Cough

Trigger: always

Action: Map Parameter ☑ Invert

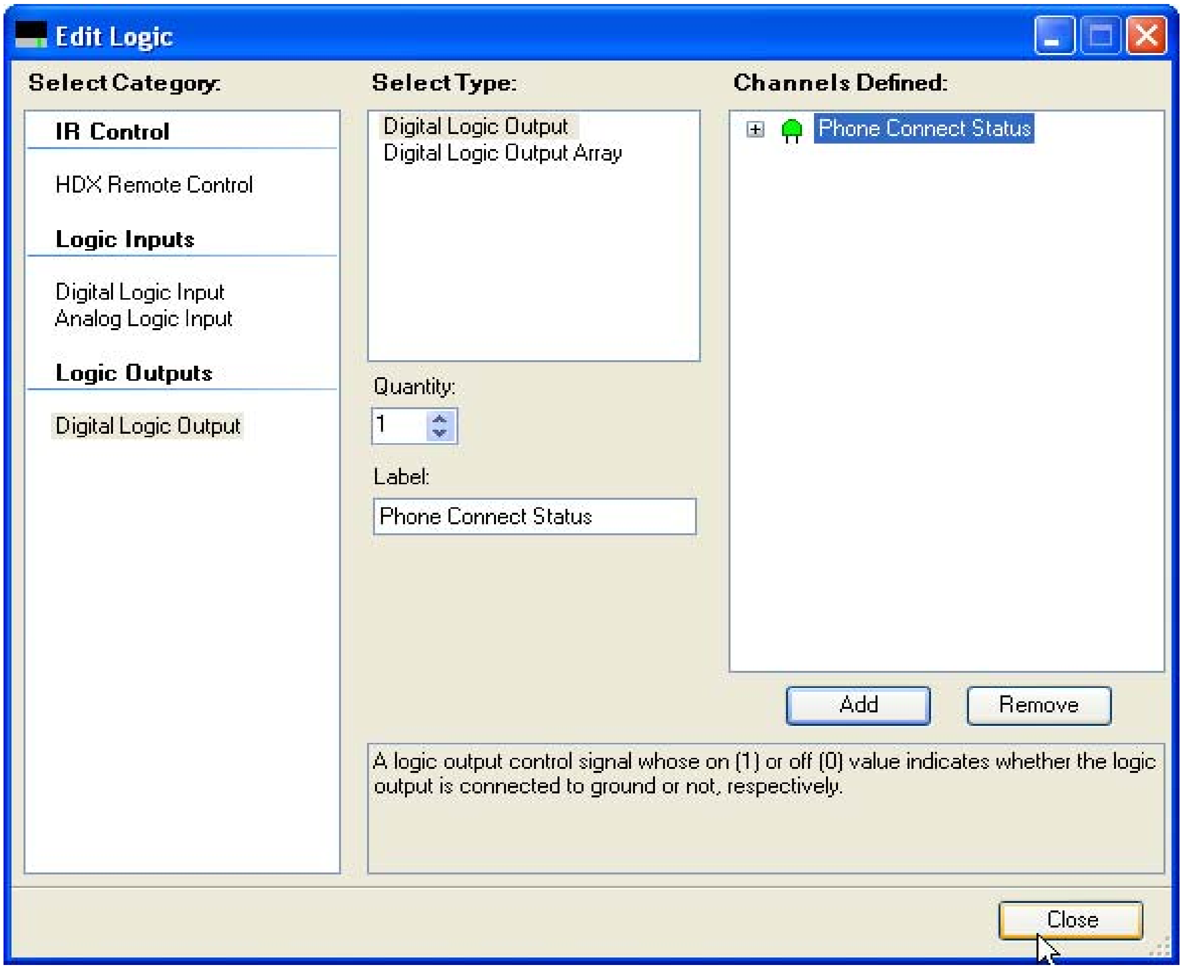
mute | Table Mic

Viewing the Phone Off Hook Drives A Relay Example

In this example, the status of the phone_connect parameter will be used to drive a logic output that is connected to a relay that can control an external circuit for illuminating a sign to indicate the phone is offhook.

Step 1: Add the logic output that will be used to drive the relay

In this example, an analog logic output called “Phone Connect Status” was created.


Edit Logic
Select Category:
IR Control
HDX Remote Control
Logic Inputs
Digital Logic Input
Analog Logic Input
Logic Outputs
Digital Logic Output
Select Type:
Digital Logic Output
Digital Logic Output Array
Channels Defined:
Phone Connect Status
Add
Remove
Close

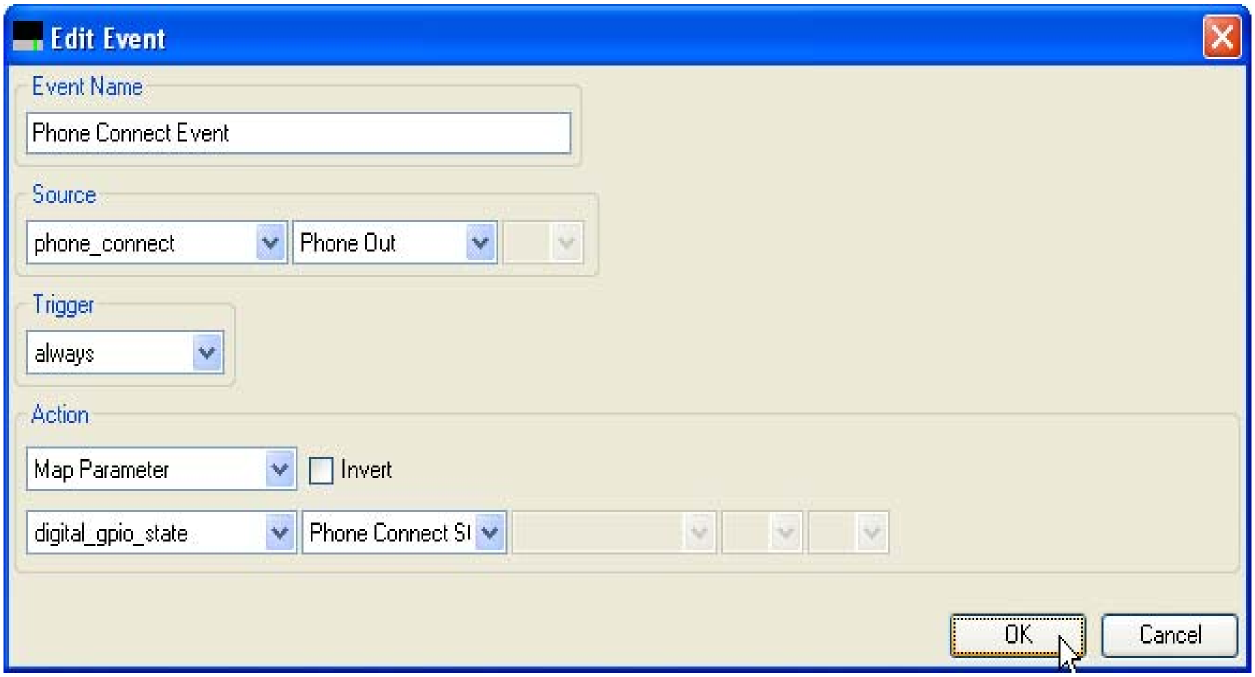
Quantity: 1

Label: Phone Connect Status

A logic output control signal whose on (1) or off (0) value indicates whether the logic output is connected to ground or not, respectively.

Step 2: Create the Event

In this example, the phone connect parameter is mapped to the logic output. If the phone is off hook (phone_connect = 1) then the logic output will allow current to flow and the relay will energize. The invert option is not necessary in this example.



Event Name: Phone Connect Event

Source: phone_connect, Phone Out

Trigger: always

Action: Map Parameter ☐ Invert

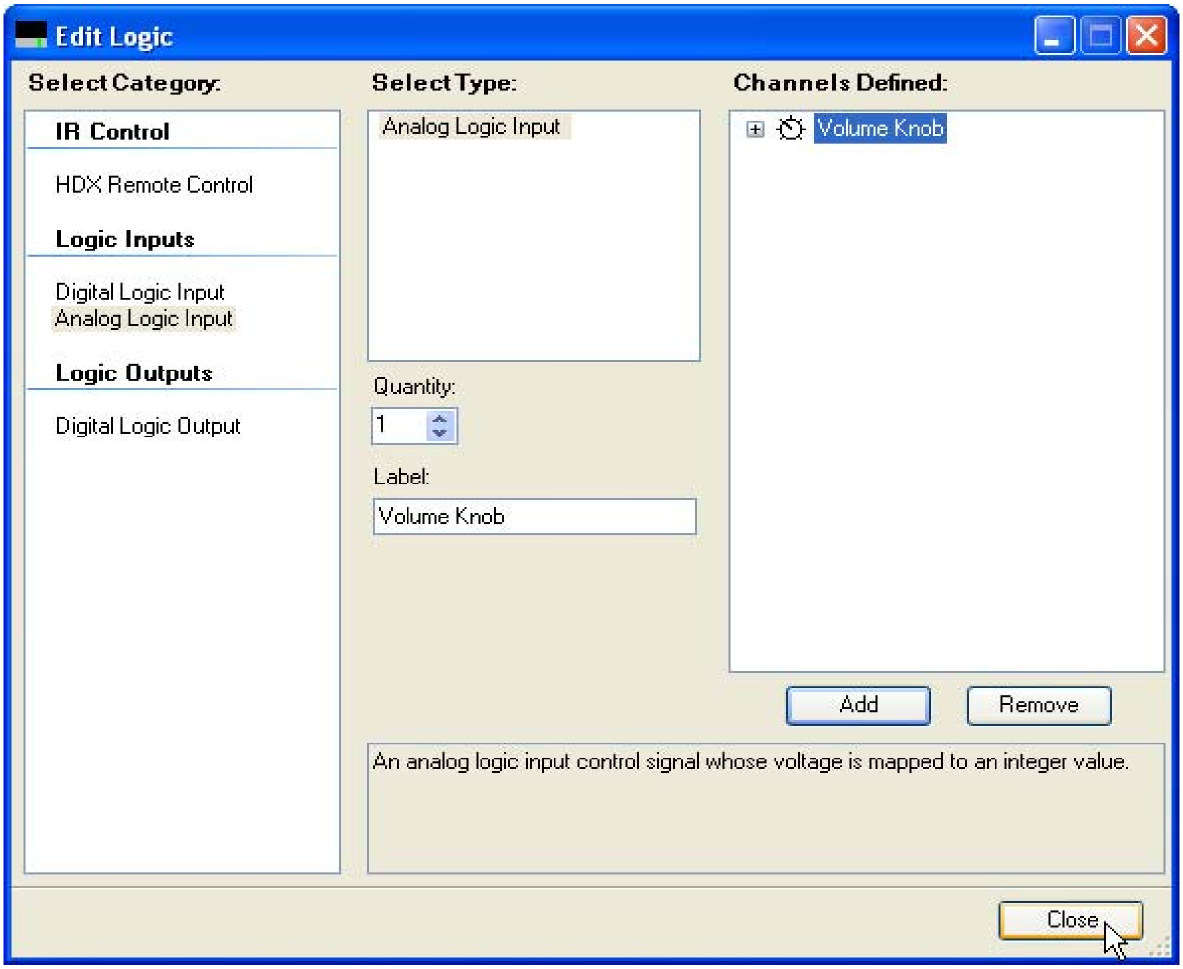
digital_gpio_state, Phone Connect St

Viewing the Volume Knob Adjusts “Amplifier” Fader Example

In this example, a volume knob will be used to control volume of an output named “Amplifier”.

Step 1: Add the analog logic input

In this example a single Analog Logic Input was created and named this logic input Volume Knob .



Edit Logic
Select Category:
IR Control
HDX Remote Control
Logic Inputs
Digital Logic Input
Analog Logic Input
Logic Outputs
Digital Logic Output
Select Type:
Analog Logic Input
Channels Defined:
Volume Knob
Add
Remove
Close

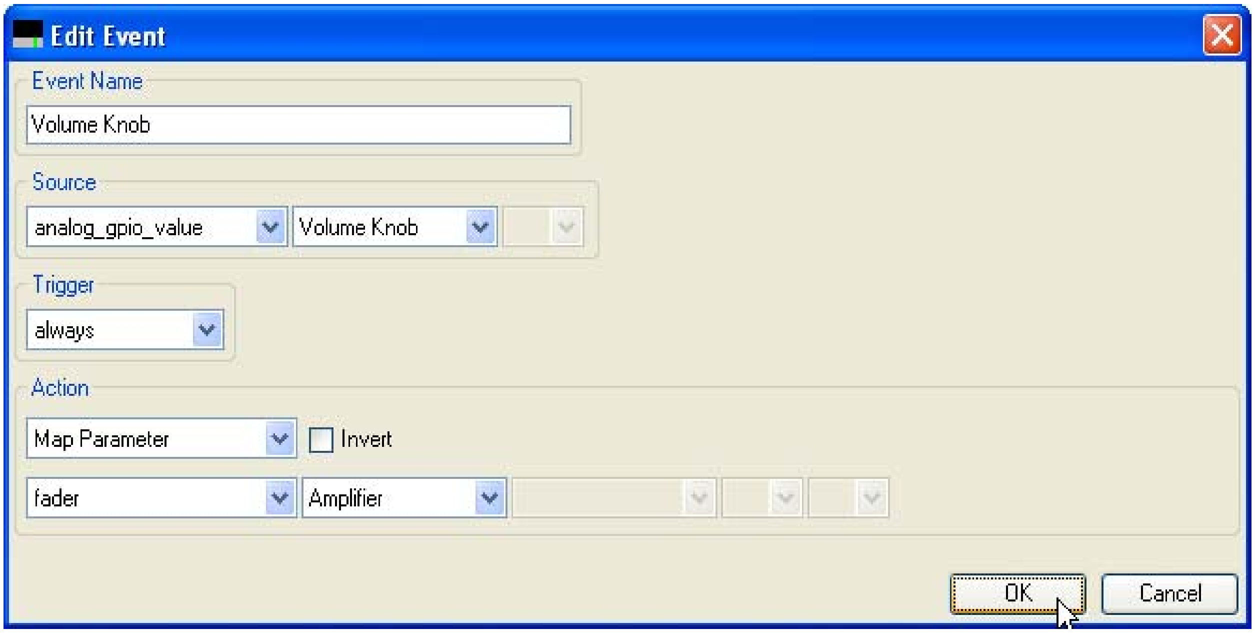
Quantity: 1

Label: Volume Knob

An analog logic input control signal whose voltage is mapped to an integer value.

Step 2: Create the event that will map the volume knob to the fader

In this example, the event maps the volume knob value to the fader of the “Amplifier” channel.



Event Name: Volume Knob

Source: analog_gpio_value, Volume Knob

Trigger: always

Action: Map Parameter ☐ Invert

fader, Amplifier

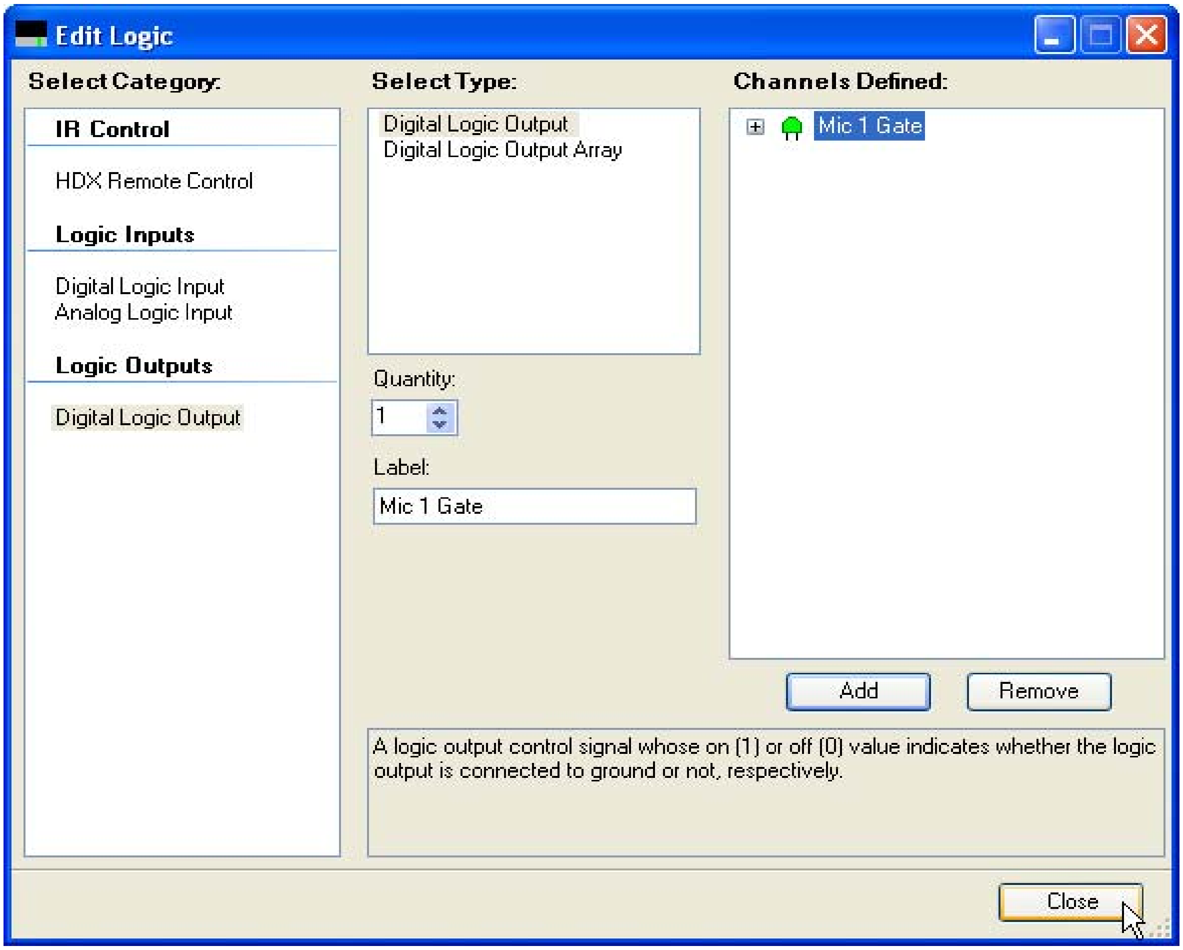
If there are user min and max fader limits set on the “Amplifier” channel, then those limits will be used automatically with the map event.

Viewing the Gating Information Sent To A Control System Example

In this example, a logic output will be used to indicate that a particular microphone has gated on. When the gate status changes, the logic output will change and the SoundStructure system will send a command acknowledgment that could be used by a control system to indicate that a microphone has gated on.

Step 1: Add the Logic Output pin

In this example, a single logic output pin called “Mic 1 Gate” is created. Check the wiring page and the logic connections to a desired logic output pin if required.


Edit Logic
Select Category:
IR Control
HDX Remote Control
Logic Inputs
Digital Logic Input
Analog Logic Input
Logic Outputs
Digital Logic Output
Select Type:
Digital Logic Output
Digital Logic Output Array
Channels Defined:
Mic 1 Gate
Add
Remove
Close

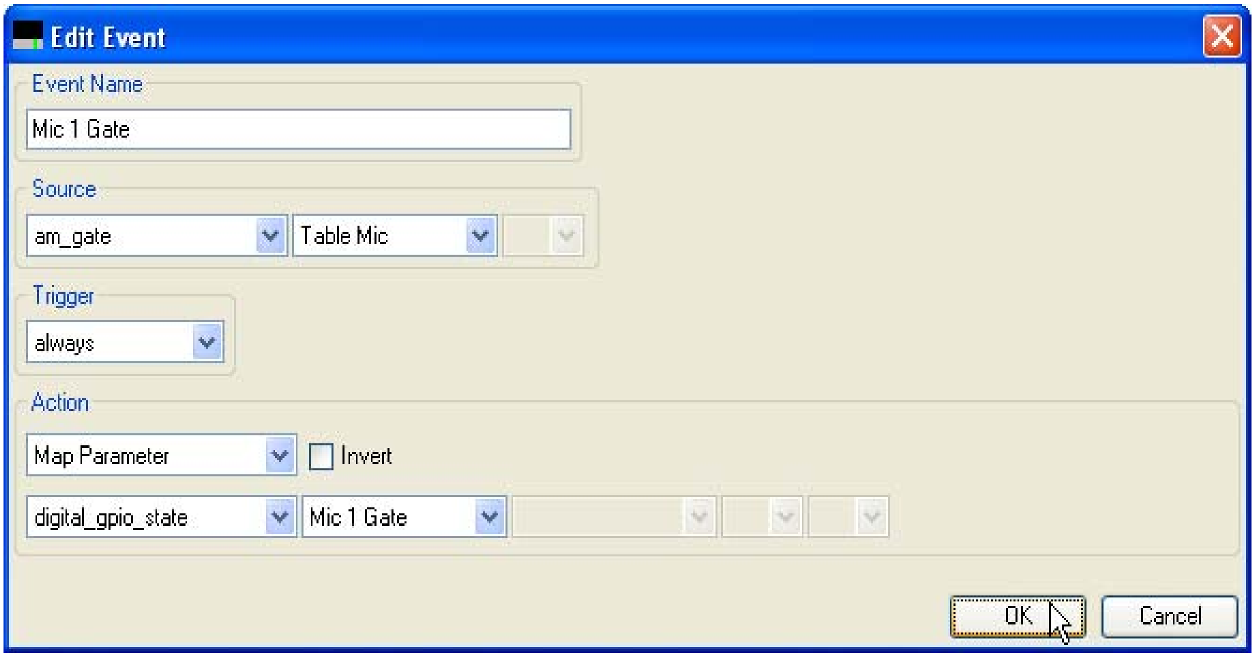
Quantity: 1

Label: Mic 1 Gate

A logic output control signal whose on (1) or off (0) value indicates whether the logic output is connected to ground or not, respectively.

Step 2: Create the event

Once the logic pin has been defined, an event mapping the microphone gating status to the logic output can be created.



Event Name: Mic 1 Gate

Source: am_gate | Table Mic

Trigger: always

Action: Map Parameter ☐ Invert

digital_gpio_state | Mic 1 Gate

In this example, when the Table Mic gates on the automixer, the logic output will be set to 1 and when the microphone gates off, the logic output will be set to 0.

When the microphone gates on, the SoundStructure system will send the acknowledgment:

val digital_gpio_state "Mic 1 Gate" 1

when the microphone gates off, the following acknowledgment will be sent:

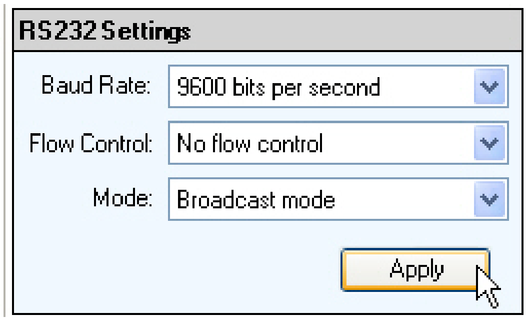
val digital_gpio_state "Mic 1 Gate" 0

A control system can use the acknowledgments from the logic output pins to indicate on a touch panel that the particular microphone is gated on or off. If there are multiple microphones in a system, each microphone can have events that connect the microphone gating status to a different logic output pin.

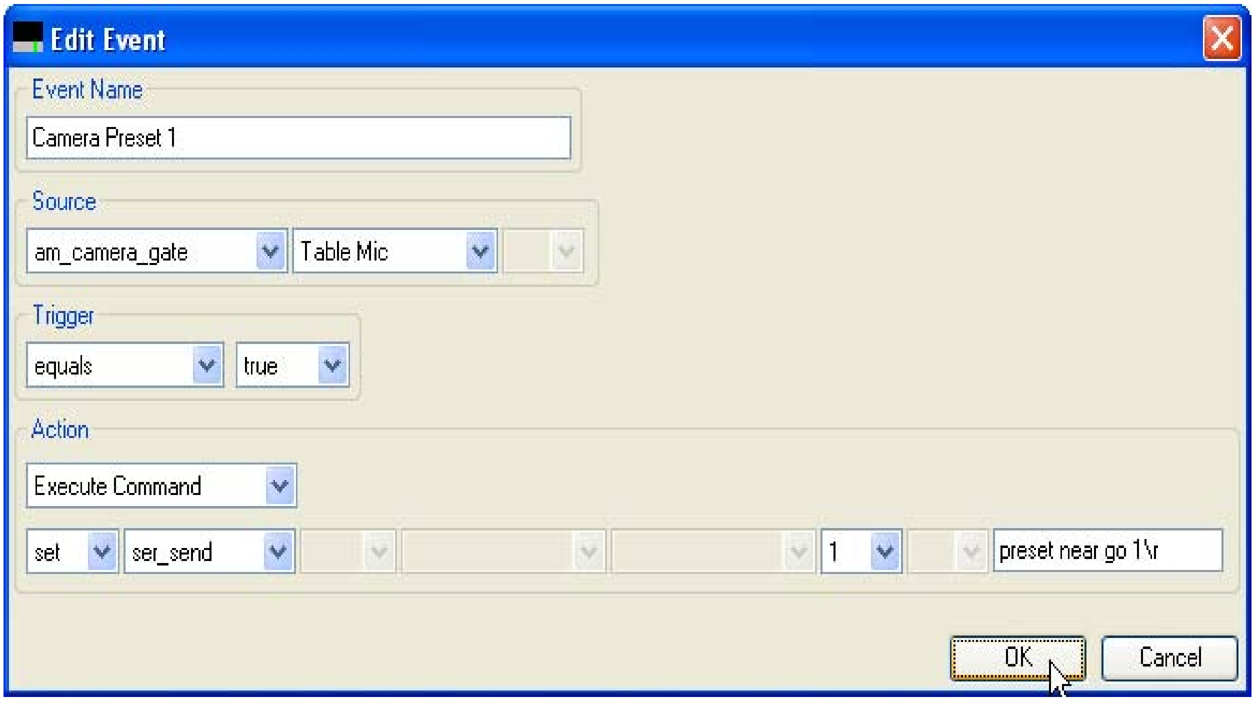
Positioning A Polycom Video Codec Camera Example

In this example, camera gating information will be used to send a command to a video codec over the serial interface of SoundStructure.

This example assumes that serial port on SoundStructure has been placed in broadcast mode. This may be configured from the wiring page as shown in the following figure.



Baud Rate: 9600 bits per second
Flow Control: No flow control
Mode: Broadcast mode



Event Name: Camera Preset 1
Source: am_camera_gate, Table Mic
Trigger: equals, true
Action: Execute Command
set ser_send 1 preset near go 1\r

This example shows that the ser_send command is used to send the serial command from the serial port on the SoundStructure with device ID 1. The command being sent is **preset near go 1**. The “\r” at the end of the command name represents a carriage return.

If an RS232 port is in the broadcast mode then that serial port cannot be used for controlling the SoundStructure system from an external control system or SoundStructure Studio.

Creating SoundStructure Events Best Practices

When creating SoundStructure events, the following recommendations will make it easier to use events.

1. Define logic inputs and outputs before trying to use logic inputs or outputs with events.

2 Confirm the wiring of the defined logic inputs and outputs matches the physical wiring to the DB25 connectors on SoundStructure devices. If necessary move logic pins definitions on the wiring page to match the physical wiring.

Note: Moving Logic Pins on the Wiring Page

If logic pins are moved on the wiring page, save the project file to ensure the settings are stored permanently into the SoundStructure device.

3 Double check the source, trigger, and action to ensure the event does what you desire.

4 Test logic inputs and events when working offline with SoundStructure Studio.

Logic Pins Forced to Open or Close

Logic input pins may be forced to closed or open with

set `digital_gpio_state “logic pin”` 0 command to close a switch or

set `digital_gpio_state “logic pin”` 1 to open a switch. The name “logic pin” should be replaced by the name of the pin you are testing.

5 Use the event enable/disable option if it is necessary to isolate and test individual events

6 Select event names that are meaningful to make it easier to interpret the event list

Managing SoundStructure Systems

This chapter describes the network and control aspects of SoundStructure systems including managing the device over IP and configuring the RS-232 port.

Connecting To The Device

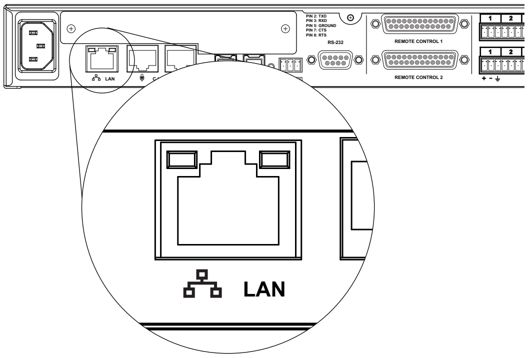
SoundStructure devices have a LAN interface and RS-232 port that may be used to configure, control, and update the system software. This section describes both the LAN and RS-232 interfaces.

When multiple devices are linked over OBAM, only one Ethernet interface or RS-232 port is required to be used, although any of the ports may be used.

LAN Interface

SoundStructure devices include a rear-panel LAN interface, shown in the following figure, that supports 10/100 Mbps communication with Auto-MDIX (medium dependent interface crossover) capability. Auto-MDIX enables the use of either a standard CAT5e cable or cross-over cable to connect to an Ethernet network. The SoundStructure device will detect either connection and work properly.

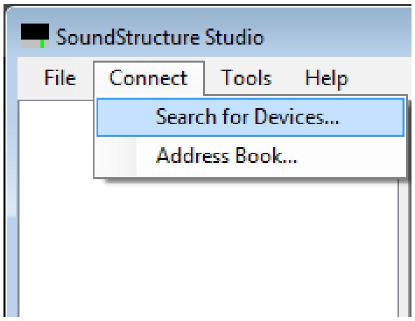
Rear Panel LAN Interface on SoundStructure Devices


LAN
RS-232
REMOTE CONTROL 1
REMOTE CONTROL 2

Dynamic IP Addresses

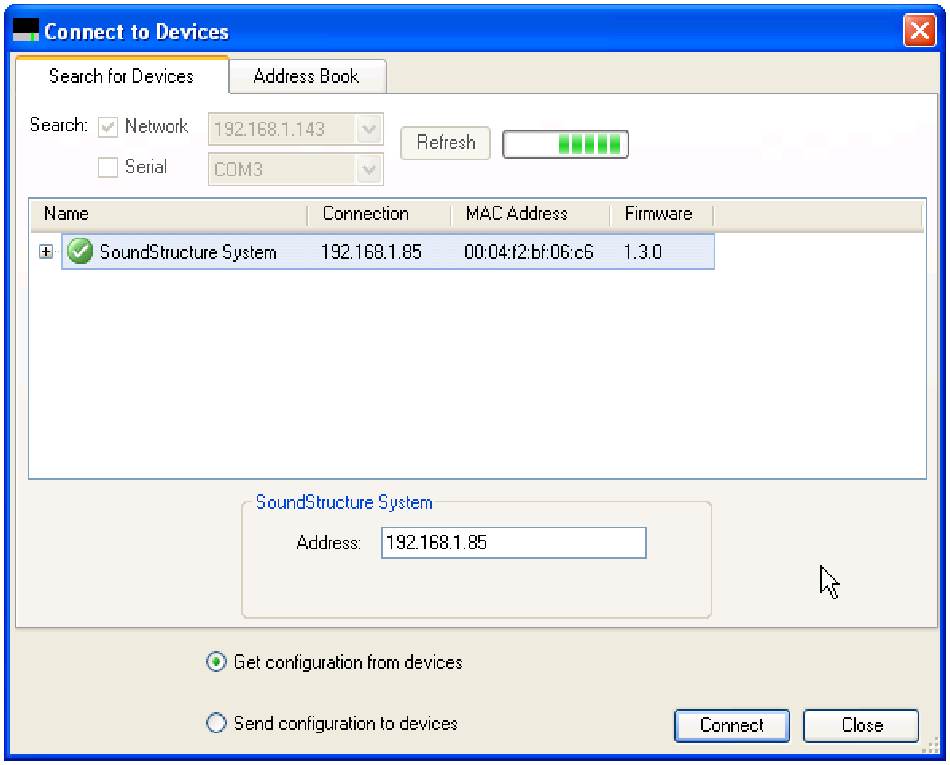
By default, the SoundStructure device accepts an IP address from a DHCP server. Once assigned, IP addresses can be determined with the SoundStructure Studio software via the SoundStructure device discovery method. To determine the IP address, connect to the device using the **Search for Devices** option as shown in the following figure.

Searching Devices in SoundStructure Studio


SoundStructure Studio
File
Connect
Tools
Help
Search for Devices...
Address Book...

SoundStructure Studio will display a list of systems found on the network interface specified by the **Search Network** option. The SoundStructure systems that are found will be shown with their system name, IP addresses, MAC addresses, and firmware version as shown in the following figure.

SoundStructure Systems Listed in SoundStructure Studio


Connect to Devices
Search for Devices
Address Book
Refresh
SoundStructure System
Connect
Close

Search: ☑ Network 192.168.1.143

☐ Serial COM3

Name	Connection	MAC Address	Firmware
SoundStructure System	192.168.1.85	00:04:f2:bf:06:c6	1.3.0

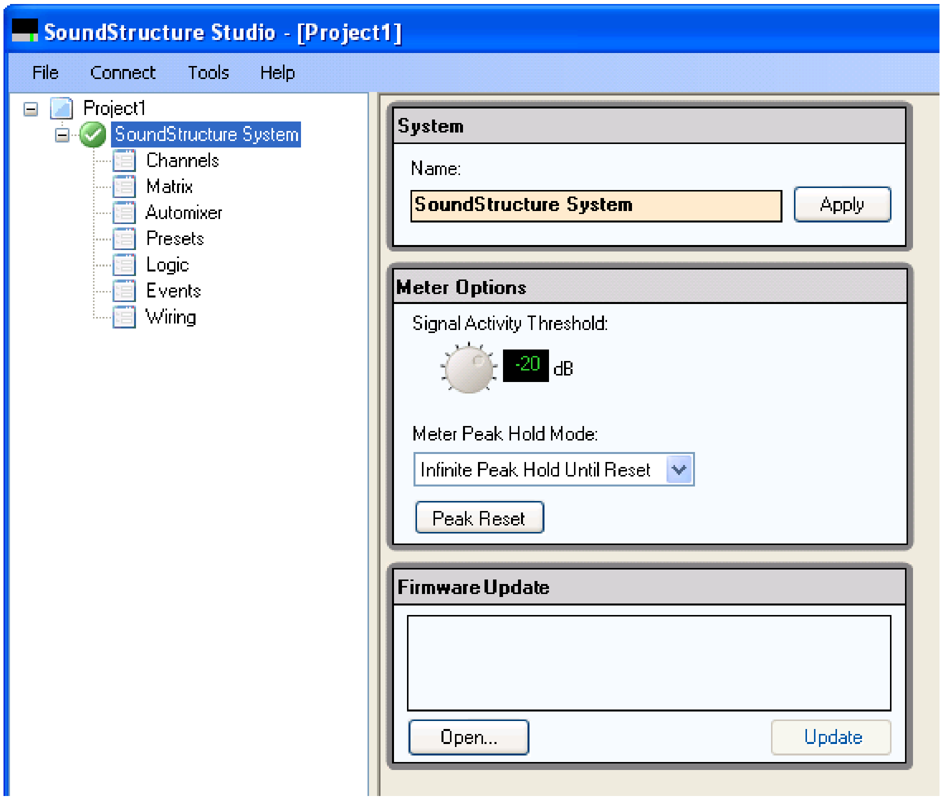
Address: 192.168.1.85

● Get configuration from devices

○ Send configuration to devices

By default the system name is set to “SoundStructure System” as shown in the next figure.

Default SoundStructure System Name


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring
System
Apply
Meter Options
Peak Reset
Firmware Update
Open...
Update

Name: SoundStructure System

Signal Activity Threshold: -20 dB

Meter Peak Hold Mode: Infinite Peak Hold Until Reset

The system name is used to easily identify units and can be set with the SoundStructure Studio as shown in the previous figure by entering the name and pressing the **Apply** button or by using the sys_name API command as shown below.

```
set sys_name “Room 475B”
```

the system will respond with the command acknowledgment

```
val sys_name “Room 475B”
```

Now the system name has been set to “Room 475B” and that’s how it will be identified during the next time **Connect to Devices** is selected. Save the project to disk with the File Save or File Save As option from SoundStructure Studio to save the file to disk and also when working online to force the SoundStructure device to store the system name permanently.

Link-Local IP Addresses

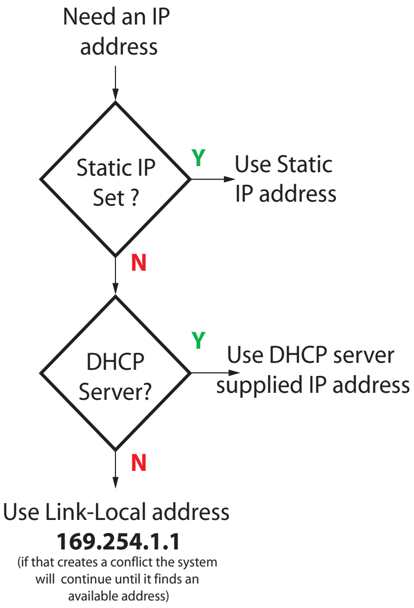
SoundStructure devices configured for DHCP and running version 1.3 or later firmware will default to the link-local IP address of 169.254.1.1 when there are no DHCP servers available to provide an IP address when the SoundStructure device powers up. The link local addressing makes it possible to connect to a computer directly to a SoundStructure device with either a straight-through or crossover Ethernet cable without either having to set a static IP address or having a DHCP server available.

Assuming the computer that is running SoundStructure Studio also does not have an IP address provided by a DHCP server, the local computer will also have a link-local address of the form 169.254.abc.def. The computer may be connected either directly to the SoundStructure device or connected through a network to the SoundStructure device, and then SoundStructure Studio will be able to automatically discover and connect to the SoundStructure device. When connected directly to the device, the local computer may use either with a straight-through or cross-over Cat5 cable.

The following figure shows how the SoundStructure device gets its IP address. If there is a static IP address assigned to SoundStructure, then that address will be used. If there is a DHCP server, then the SoundStructure device will use the address provided by the DHCP server. If there is no DHCP server, then the SoundStructure device will use a locally generated link-local IP address which will default to 169.254.1.1 assuming this does not create a conflict with a different device on the network.

Please note that when the SoundStructure device has a link-local IP address, if a DHCP server comes online at a later time, the SoundStructure device will accept an IP address provided by the DHCP server and will no longer have the link-local IP address.

SoundStructure IP Address


Need an IP address
Static IP Set ?
Y
Use Static IP address
N
DHCP Server?
Y
Use DHCP server supplied IP address
N
Use Link-Local address
169.254.1.1
(if that creates a conflict the system will continue until it finds an available address)

Static IP Addresses

SoundStructure devices may also be assigned a static IP address directly from SoundStructure Studio or manually via the API and a terminal session.

Assigning A Static IP Address Via Ethernet

Over the network interface, first connect the device to the network and accept the dynamic IP address from a DHCP server. Once connected to the device, the static IP address may be set directly from the wiring page within SoundStructure Studio. Hit the apply button to change the IP address. Changing the IP address will force SoundStructure Studio to lose connection to the device.

Reconnect to the SoundStructure device using the new IP address and save the settings to a file using the File Save or File Save As which when working online will also force the SoundStructure device to store the IP address permanently in the device. The project must be saved to ensure the IP address is set permanently - otherwise upon the next reboot of the device, the IP address will revert to the previous settings.

With version 1.3 firmware and later, a static IP address assigned to a device will remain with the SoundStructure device regardless of whether a configuration file in a device matches that actual hardware. In previous firmware releases, if the configuration file did not match the hardware, perhaps because a telephony card was inserted or removed from a SoundStructure device, then the system would default back to looking for a DHCP address.

Assigning A Static IP Address Via The API

A terminal window may be opened directly via the RS-232 interface to send the API commands described below.

To configure the device to have a static IP address, use the eth_settings command as follows:

```
set eth_settings 1
“mode='static',addr='192.168.1.101',gw='192.168.10.254',nm='255.255.255.0',dns='66.82.
134.56'”
```

where the 1 represents the device ID of the SoundStructure. If multiple SoundStructure devices are linked together, the device id of the first unit will be 1 and subsequent connected devices will have sequential device id’s ranging from 2 to the number of devices assuming the OBAM interface is connected from OBAM in to OBAM out as described in Introducing the Polycom SoundStructure Product Family and Introducing SoundStructure Design Concepts.

The `eth_settings` command accepts a complete string with the arguments summarized below. Arguments to the different fields must be surrounded with single quotes.

eth_settings Commands and Values

Field	Definition	Values
mode	How the system receives an IP address	static or dhcp
addr	IP address	The desired IP address
gw	Gateway	The IP address of the gateway
nm	Netmask	The netmask defining the subnet
dns	Name Server	The IP address of the name server used to resolve host names. Multiple DNS servers may be specified by separating the arguments with spaces

If the mode is set to ‘*dhcp’* then the remaining arguments are accepted but not used until the mode is set to static. All arguments have to be sent if the address is being set to a static IP address.

To enable SoundStructure devices to accept a dynamic IP address use the command:

```
set eth_settings 1 “type='dhcp'”
```

where 1 represents the default device ID of a stand-alone SoundStructure device. Please note that there are single quotes around the argument ‘dhcp’ and the entire argument string is enclosed in double quotes.

To query the IP settings of the device, use the get action as in the following example:

```
get eth_settings 1
val eth_settings 1 “mode='dhcp',addr='172.22.2.110',dns='172.22.1.1 172.22.1.2',
gw='172.22.2.254',nm='255.255.255.0'"
```

To set the address to a static IP address, follow this example:

```
set eth_settings 1 “mode='static',addr='172.22.2.110',dns='172.22.1.1 172.22.1.2',
gw='172.22.2.254',nm='255.255.255.0'"
```

All the arguments to the eth_settings command must be specified when the mode is set to ‘static’.

Once the IP address settings have been changed, it is important to make sure that the project file settings are saved to disk or a preset is saved because this will ensure the IP address of the SoundStructure device is written permanently to the SoundStructure’s non-volatile memory. Failure to save the file or save settings to a preset will mean the IP address will revert back to the previously permanently saved IP address upon a power cycle.

Note: Reconnecting SoundStructure Devices when IP Address Changes

If connected via IP to the SoundStructure device and the IP address is changed, *reconnect* to the system using the new IP address and either save the settings to a file or save the settings to a preset to ensure the new IP address is stored permanently in the SoundStructure devices.

Setting The Time Server

To set the time server, use the command ntp_server as shown in the example below:

```
set dev_ntp_server 1 “pool.ntp.org”
val dev_ntp_server 1 “pool.ntp.org”
```

where 1 is the device ID of the SoundStructure. See Appendix A for more information on API commands associated with the Ethernet interface.

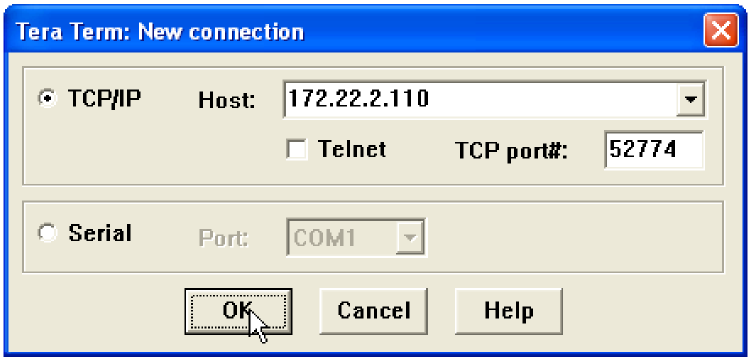
Control And Command Sessions

SoundStructure systems are controlled and configured with the SoundStructure API command set via a communication to port **52774**.

The number of active TCP control sessions on port 52774 is unlimited subject to network bandwidth to access the device.

The following figure shows the initiation of a TCP/IP connection to a SoundStructure device at address 172.22.2.110 and port 52774 using a third party terminal program.

Initiation of TCP/IP Connection to SoundStructure Device



● TCP/IP Host: 172.22.2.110

☐ Telnet TCP port#: 52774

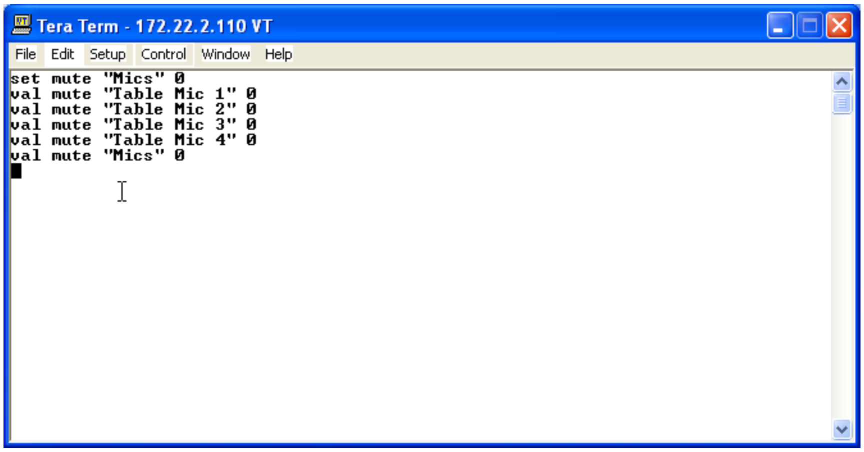
○ Serial Port: COM1

Once the control session has been initiated, commands may be sent to the device and command acknowledgments received as shown in the following figure where a mute command is sent to the virtual channel group “Mics”.

```
set mute “Mics” 0
```

The command responses are received back and include the mute status for all virtual channels in the “Mics” virtual channel group.

Received Command Responses



```
set mute "Mics" 0
val mute "Table Mic 1" 0
val mute "Table Mic 2" 0
val mute "Table Mic 3" 0
val mute "Table Mic 4" 0
val mute "Mics" 0
```

When there are multiple simultaneous control sessions to a SoundStructure system, the control session that sends commands will also receive command acknowledgments for all of its commands. Other control sessions will only receive command acknowledgments from a command entered from another control session if a parameter value changes.

For example, if a control session queries the value of the mute status, only that control session will receive the acknowledgment of the mute value. However if the control session changes the mute state, for example, all control sessions will receive an acknowledgment.

Note: Received Acknowledgments for Control Sessions

Control sessions receive acknowledgments for commands entered in that session and *only* receive command acknowledgments from other command sessions if the other command sessions change the value of a parameter.

SoundStructure Device Discovery

SoundStructure Studio uses a discovery mechanism for identifying SoundStructure devices on the network. SoundStructure Studio sends a UDP discovery broadcast using port 52774 and all SoundStructure systems that receive the broadcast will respond and identify themselves.

If the IP address changes on the SoundStructure device, such as if the dynamic IP address lease expires and a new IP address is received, it may take up to 75 seconds for the discovery mechanism to restart.

This discovery mechanism only creates network traffic when SoundStructure Studio is discovering devices caused by the user opening the Connect to Devices window.

Because both the discovery and command channels use port 52774, traversing firewalls only requires opening port 52774 for both UDP (for discovery) and TCP (for commands) to allow for remote access of the SoundStructure device.

Depending on the network router configurations in the network, SoundStructure device discovery may not work across different subnets. However it is still possible to remotely configure SoundStructure devices if the IP address of the device is known as the IP address may be typed in directly in the **Connect to Devices** user interface.

AMX Beacon

The SoundStructure devices comply with the AMX Dynamic Discovery Protocol and send a UDP broadcast to multi-cast address 239.255.250.250 port 9131 at random intervals between 30 to 60 seconds.

The broadcast beacon depends on the particular SoundStructure device model and is formatted as shown below.

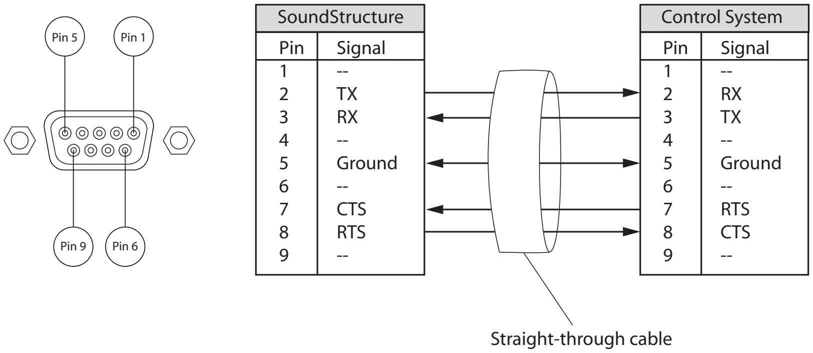
```
AMXB<-UUID=001122334455><-SDKClass=AudioConferencer><-Make=Polycom>
<-Model=SoundStructureC16><-Revision=1.0.0><Config-Name=SoundStructure C16
Configuration> <Config-URL=http://172.22.2.109/>
```

where 001122334455 is the MAC address of the SoundStructure C16 device in this example.

RS-232

The RS-232 interface is capable of running up to 115,200 bps and has a default rate of 9,600 bps, eight data bits, no parity, one stop bit (8-N-1). The pinout of the connection and the recommended straight-through cabling to a control system is shown in the following figure.

Pinout Connection and Control System Recommended Cabling



SoundStructure

Pin	Signal
1	--
2	TX
3	RX
4	--
5	Ground
6	--
7	CTS
8	RTS
9	--

Control System

Pin	Signal
1	--
2	RX
3	TX
4	--
5	Ground
6	--
7	RTS
8	CTS
9	--

The settings of the RS-232 port may be changed with the ser_baud and ser_flow settings as follows:

```
set ser_baud 1 38400
```

sets the RS-232 baud rate to 38400 bps. See Appendix A for additional information concerning the RS-232 commands.

The RS-232 port may be used for control sessions or for configuration with SoundStructure Studio.

Configuring And Accessing The Logs

The SoundStructure device logs include the following information and may be retrieved from the device using SoundStructure Studio.

1. API commands
2. API command responses
3. Error messages

The typical log will look like the following file.

```
Aug 29 14:06:05 gcp: ack: [all] val mute "Table Mic 1" 0
Aug 29 14:06:07 gcp: ack: [all] val mute "Table Mic 2" 0
Aug 29 14:06:07 gcp: ack: [all] val mute "Table Mic 3" 0
Aug 29 14:06:07 gcp: ack: [all] val mute "Table Mic 4" 0
Aug 29 14:06:07 gcp: ack: [all] val mute "Mics" 0
Aug 29 14:06:18 gcp: cmd: [172.22.2.117:1462] set matrix_mute "Mics" "Phone Out" 1
Aug 29 14:06:18 gcp: ack: [all] val matrix_mute "Table Mic 1" "Phone Out" 1
Aug 29 14:06:18 gcp: ack: [all] val matrix_mute "Table Mic 2" "Phone Out" 1
Aug 29 14:06:18 gcp: ack: [all] val matrix_mute "Table Mic 3" "Phone Out" 1
Aug 29 14:06:18 gcp: ack: [all] val matrix_mute "Table Mic 4" "Phone Out" 1
Aug 29 14:06:18 gcp: ack: [all] val matrix_mute "Mics" "Phone Out" 1
Aug 29 14:06:19 gcp: cmd: [172.22.2.117:1462] set matrix_mute "Mics" "Phone Out" 0
Aug 29 14:06:19 gcp: ack: [all] val matrix_mute "Table Mic 1" "Phone Out" 0
Aug 29 14:06:19 gcp: ack: [all] val matrix_mute "Table Mic 2" "Phone Out" 0
Aug 29 14:06:19 gcp: ack: [all] val matrix_mute "Table Mic 3" "Phone Out" 0
Aug 29 14:06:19 gcp: ack: [all] val matrix_mute "Table Mic 4" "Phone Out" 0
Aug 29 14:06:19 gcp: ack: [all] val matrix_mute "Mics" "Phone Out" 0
```

API commands correspond to the commands that were sent to the system and how they were transmitted, IP or RS-232. API command responses show the command acknowledgment and where the response was directed.

Integrating The Polycom Touch Control with SoundStructure Systems

This chapter describes how to integrate the Polycom Touch Control with a SoundStructure system. For information about using the Polycom Touch Control with a Video Codec system, see the Polycom HDX Systems Administrator's Guide.

Polycom Touch Control and SoundStructure Systems

The Polycom Touch Control is an easy-to-use touch sensitive user-interface device that integrates directly with a SoundStructure system for control of the audio conferencing system including dialing SoundStructure telephony interfaces, muting audio, and adjusting volume. SoundStructure events are used to customize the behavior when muting and adjusting volume.

The Polycom Touch Control can also integrate directly with a Polycom Video Codec system. When controlling an Video Codec system, a SoundStructure system connected to the Video Codec system over Conference Link2 will be controlled indirectly via the Conference Link2 integration. See Connecting Over Conference Link2 for additional information about Conference Link2.

SoundStructure System Requirements

SoundStructure Firmware

To use the Polycom Touch Control with the SoundStructure system that includes TEL1 or TEL2 telephony interface cards, the SoundStructure system must have firmware version 1.3.3 or later.

To use the Polycom Touch Control with a SoundStructure system that includes the SoundStructure VoIP Interface, the SoundStructure system must have firmware version 1.5.0 or later.

SoundStructure Studio

To use the Polycom Touch Control with the SoundStructure system, the SoundStructure Studio version must be 1.5.0 or later. As described in this chapter, SoundStructure Studio 1.5.0 automatically creates the necessary events for integrating the Polycom Touch Control with the SoundStructure system.

Polycom Touch Control

To use the Polycom Touch Control with the SoundStructure VoIP Interface, the Polycom Touch Control software must be version 1.4.0 or later.

The Polycom Touch Control software version 1.4.x does not operate with a SoundStructure system that has authentication enabled. To pair the Polycom Touch Control with the SoundStructure system, set the SoundStructure authentication mode to ‘open’ with SoundStructure Studio.

Polycom Touch Control with Video CodecVideo Codec Systems vs. SoundStructure Systems

To use the Polycom Touch Control, it must be first paired with the system to be controlled. A Polycom Touch Control may be paired either with:

- A Video Codec system for video conferencing applications (which may include optional SoundStructure devices), or
- A SoundStructure system for audio conferencing applications.

It is important to understand the operational differences of the Polycom Touch Control when paired with an Video Codec system vs. when paired with a SoundStructure system. An overview of these operational differences is shown in the following table.

Operational Differences for Polycom Touch Control

Polycom Touch Control	Paired with the Video Codec (+ optional SoundStructure system)	Paired with SoundStructure system
What is controlled	Controls the Video Codec directly and an optional SoundStructure system is indirectly controlled with Conference Link2 messages.	Controls the SoundStructure system via the SoundStructure API
Dials	Video Codec calls, Codec telephony interfaces	SoundStructure telephony interfaces via the SoundStructure API commands.
Volume control	Controls the volume in the Video Codec. If optional SoundStructure devices are connected over Conference Link2, SoundStructure will receive `clink_volume` messages from the Video Codec over Conference Link2.	Polycom Touch Control sends `clink_volume` API command directly to the SoundStructure system in the form of: `set clink_volume 1 N` where N ranges from 0 to 51.
Mute control	Controls the microphone mute in the Video Codec. If optional SoundStructure devices are connected over Conference Link2, SoundStructure will receive `clink_mute` messages from the Video Codec over Conference Link2.	Polycom Touch Control sends `clink_mute` API command directly to SoundStructure system in the form of: `set clink_mute 1 1` to mute `set clink_mute 1 0` to unmute

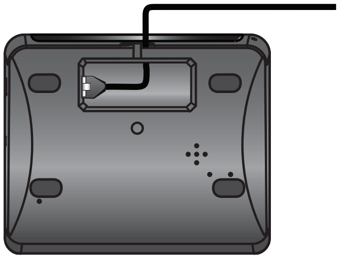
Note: Dialing SoundStructure Telephony Interfaces

To dial SoundStructure telephony interfaces, the Polycom Touch Control must be paired with the SoundStructure system.

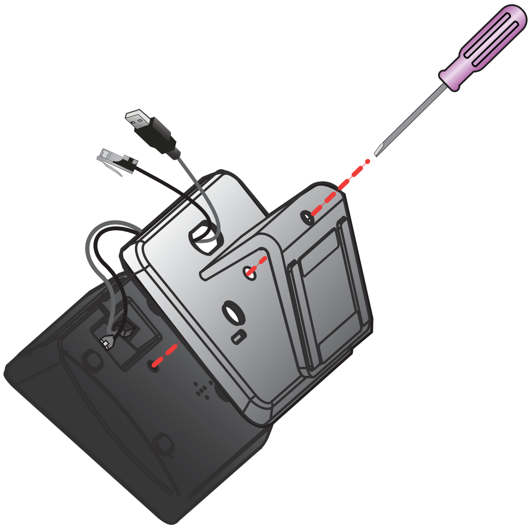
Pairing the Polycom Touch Control with SoundStructure

Preparing the Polycom Touch Control Device:

1 Connect the Ethernet cable to the underside of the Polycom Touch Control device. The Polycom Touch Control, by default, expects to receive an IP address from a DHCP server on the network. To set a static IP address on the Polycom Touch Controller, see Configuring the Polycom Touch Control LAN Properties.

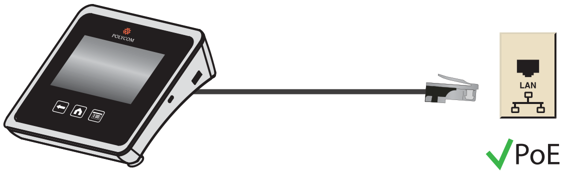


2 To use the stand, route the Ethernet cables through the opening in the stand. Then attach the stand to the Polycom Touch Control device by tightening the mounting screw with a screwdriver.

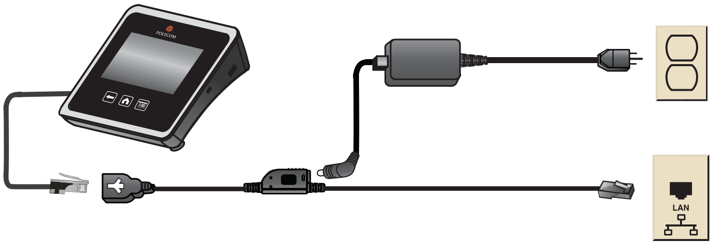


3 Plug the Ethernet cable into the wall Ethernet outlet.

- If the room provides Power Over Ethernet, connect the Ethernet cable directly to a LAN outlet as shown in the following figure.


POLYCOM
LAN
✓PoE

- If the room does not provide Power Over Ethernet, connect the Ethernet cable to the power supply adapter. Then connect the power supply adapter to a LAN outlet and power outlet as shown in the following figure.

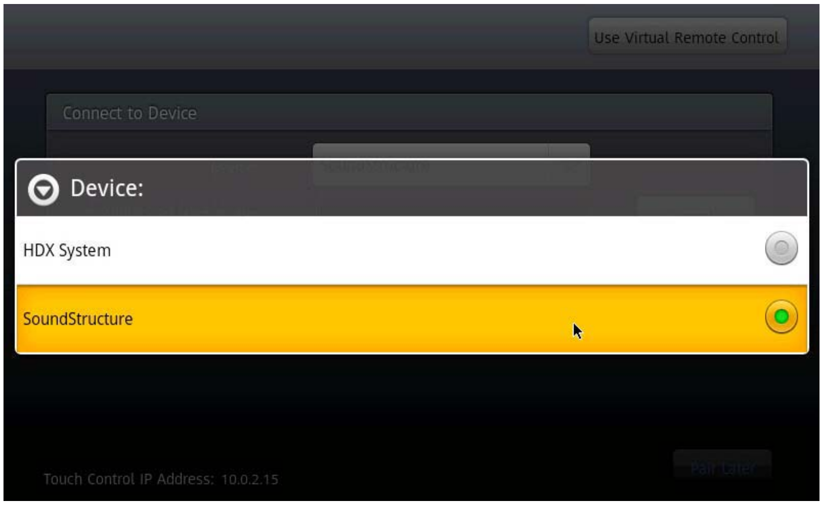

POLYCOM
LAN

The Polycom Touch Control device powers on and displays the language selection screen.

4 Choose the desired language and follow the on-screen instructions to pair the Polycom Touch Control device with the SoundStructure system.

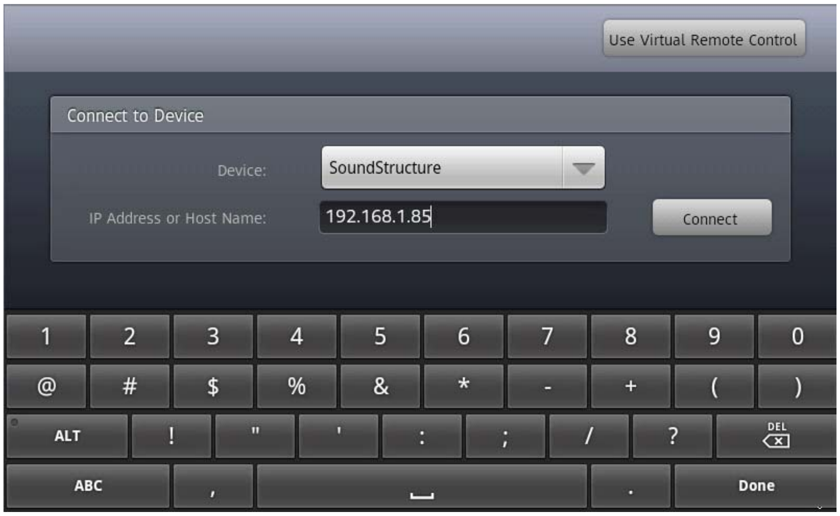
To Pair the Polycom Touch Control with the SoundStructure System

1 Select SoundStructure from the Connect to Device menu as shown in the following figure.


Use Virtual Remote Control
Connect to Device
Device:
HDX System ○
SoundStructure ●
Pair Later

Touch Control IP Address: 10.0.2.15

2 Enter the IP address of the desired SoundStructure system as shown in the following figures. A keyboard will appear once the IP address field is touched as shown below.

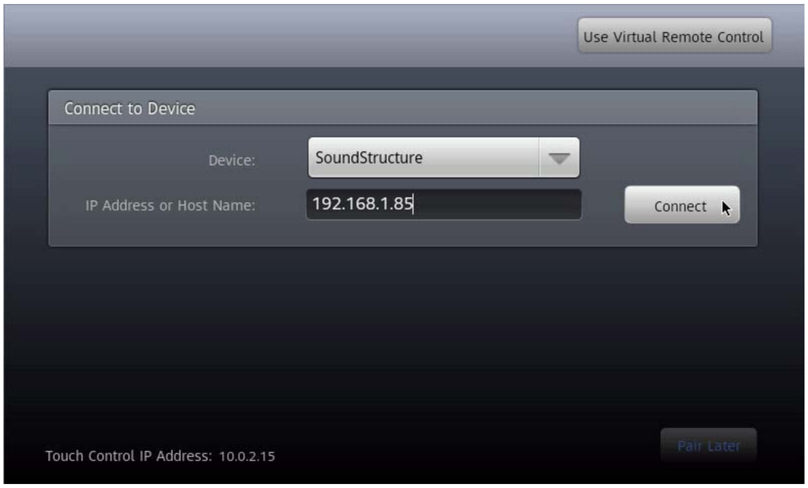

Connect to Device

Device: SoundStructure

IP Address or Host Name: 192.168.1.85

Use Virtual Remote Control
Connect
1 2 3 4 5 6 7 8 9 0
@ # $ % & * - + ( )
ALT ! " ' : ; / ? DEL
ABC , . Done

If the IP address of the SoundStructure system is not known, use SoundStructure Studio to discover the IP address of the SoundStructure system.


Connect to Device

Device: SoundStructure

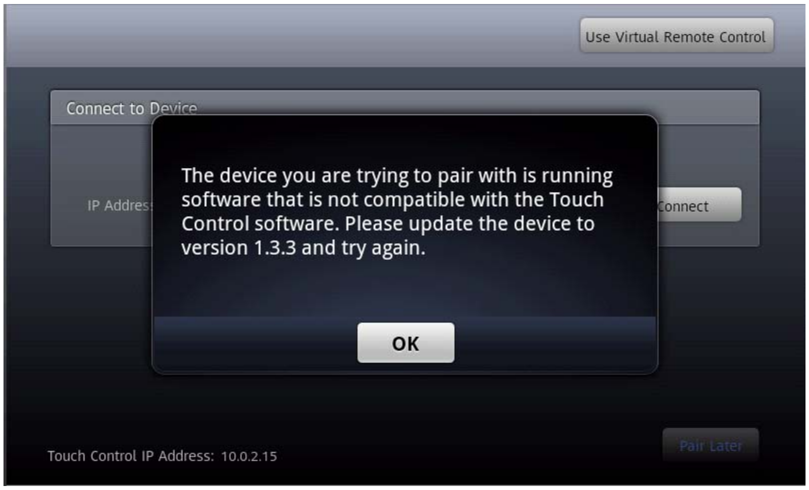
IP Address or Host Name: 192.168.1.85

Touch Control IP Address: 10.0.2.15

Use Virtual Remote Control
Connect
Pair Later

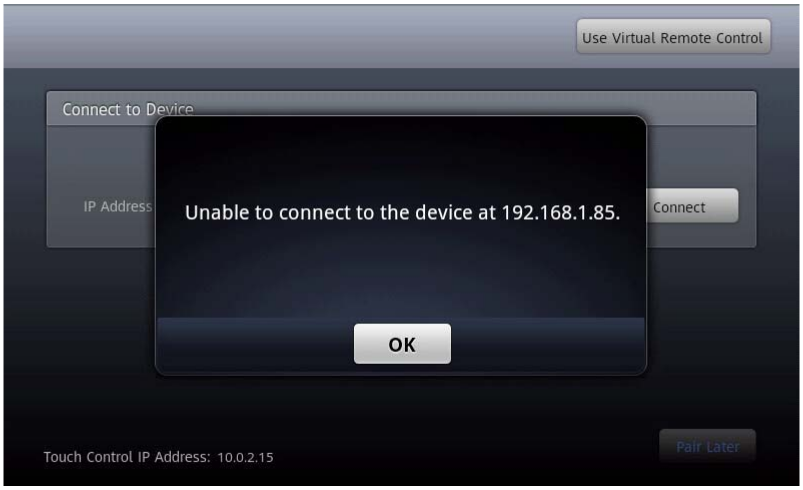
3 Press Connect to initiate pairing.

4 If the system does not pair successfully because the firmware in the SoundStructure system is older than version 1.3.3 than the system will display an error message as shown in the following figure. To resolve this issue, update the firmware in the SoundStructure system to at least version 1.3.3.


The device you are trying to pair with is running software that is not compatible with the Touch Control software. Please update the device to version 1.3.3 and try again.

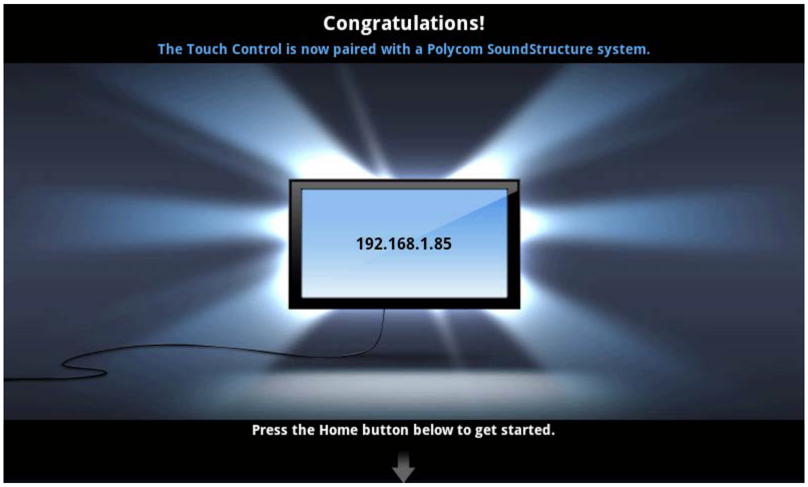
Touch Control IP Address: 10.0.2.15
Use Virtual Remote Control
Connect to Device
IP Address
Connect
OK
Pair Later

5 If the SoundStructure system firmware is at least version 1.3.3 and system does not pair successfully, the Polycom Touch Control will display a screen as shown in the following figure. At this point, it will be necessary to confirm the IP address of the SoundStructure system, and the confirm that the Polycom Touch Control has a valid IP address and that there is a network route to the SoundStructure system and then press Connect again.


Unable to connect to the device at 192.168.1.85.

Touch Control IP Address: 10.0.2.15
Use Virtual Remote Control
Connect to Device
IP Address
Connect
OK
Pair Later

6 Once the system pairs successfully, the Polycom Touch Control will display a successful pairing screen as shown in the following figure.



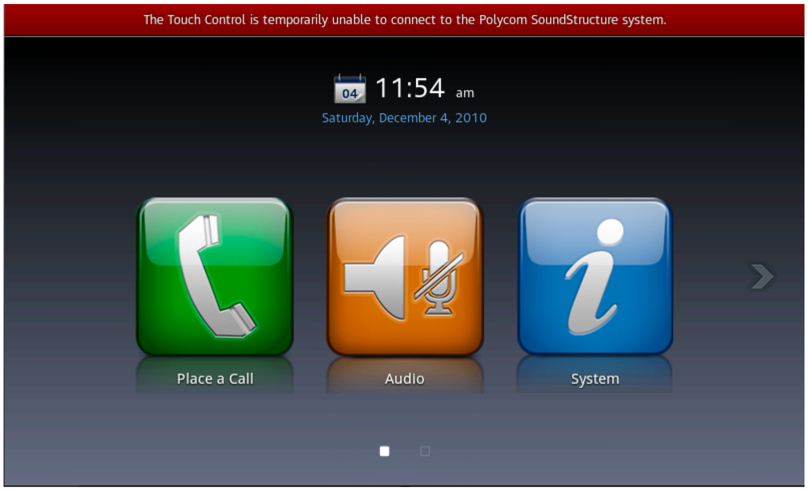
Congratulations!

The Touch Control is now paired with a Polycom SoundStructure system.

Press the Home button below to get started.

If the network connection is lost for any reason, the Polycom Touch Control device automatically attempts to restore the connection.

If the connection is lost the Polycom Touch Control will show a banner across the top of the screen that indicates the connection to the SoundStructure has been temporarily lost as shown in the following figure. If this message appears, then check that there is a valid network connection between the Polycom Touch Control and that the SoundStructure is powered on.



The Touch Control is temporarily unable to connect to the Polycom SoundStructure system.

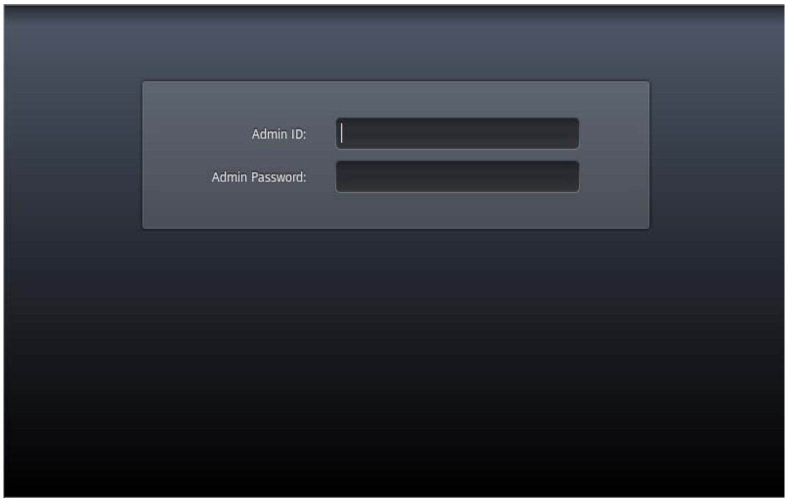
Polycom Touch Control Administrative Settings

The Polycom Touch Control device has separate administration settings that allow for updating the Polycom Touch Control software and configuring LAN, regional, and security properties for the Polycom Touch Control.

To access Admin settings, touch the screen and move from right to left to access the Administration icon 🛠.

The Administration page requires a login and password, as shown in the figure below. The default login is **Admin** and the default password is **456**.

. Administration Login Page



Configuring the Polycom Touch Control LAN Properties

To configure Polycom Touch Control LAN settings:

1. From the Home screen, touch 🛠.
2. Touch the **LAN Properties** tab.
3. Configure the following **IP Address (IPv4)** settings:

IP Address Settings

Setting	Description
Set IP Address	Specifies how the Polycom Touch Control obtains an IP address. • **Obtain IP address automatically** — Select if the Touch Control gets an IP address from the DHCP server on the LAN. • **Enter IP address manually** — Select if the IP address is not automatically assigned.
IP Address	If the Polycom Touch Control obtains its IP address automatically, this area displays the IP address currently assigned to the Polycom Touch Control. If you selected **Enter IP address manually**, enter the IP address here.

IP Address Settings

Setting	Description
Subnet Mask	Displays the subnet mask currently assigned to the Polycom Touch Control. If you selected **Enter IP address manually**, enter the subnet mask here.
Default Gateway	Displays the gateway currently assigned to the Polycom Touch Control. If you selected **Enter IP address manually**, enter the gateway IP address here

4 Configure the following DNS settings:

DNS Setting

Setting	Description
Domain Name	Displays the domain name currently assigned to the Polycom Touch Control. If the Polycom Touch Control does not automatically obtain a domain name, enter one here.
DNS Servers	Displays the DNS servers currently assigned to the Polycom Touch Control. If the Polycom Touch Control does not automatically obtain a DNS server address, enter up to two DNS servers here. You can specify IPv4 DNS server addresses only when the IPv4 address is entered manually. When the IPv4 address is obtained automatically, the DNS Server addresses are also obtained automatically.

Configuring Polycom Touch Control Regional Settings

To configure the Polycom Touch Control Regional settings:

1. From the Home screen, touch 🛠.
2. Touch the **Location** tab.
3. Select a language from the **Language** drop-down menu.
4. Set the Date and Time information as described in the following table.

Time Information Settings

Setting	Description
Time Zone	Specifies the time difference between GMT (Greenwich Mean Time) and your location.
Time Server	Specifies connection to a time server for automatic Touch Control time settings. The date and time must be manually reset every time the Touch Control restarts, in the following cases: • **Time Server** is set to **Off** • **Time Server** is set to **Manual** or **Auto**, but the Touch Control cannot connect to a time server successfully.

Time Information Settings

Setting	Description
Time Server Address	Specifies the address of the time server to use when **Time Server** is set to **Manual**.
Time Format	Specifies your format preference for the time display and lets you enter your local time.

Configuring Security Options

Configuring Admin ID and Password for Polycom Touch Control device

It is possible to set an Admin ID and password, which limits access to the Polycom Touch Control Administration settings.

1. From the Home screen touch 🛠.
2. Touch the **Security** tab.
3. Set the following security settings:

Security Setting

Setting	Description
Admin ID	Specifies the ID for the administrator account. The default Admin ID is “admin”.
Admin Password	Specifies the password for administrator access when logging in to the Polycom Touch Control. When this password is set, you must enter it to configure the Polycom Touch Control Admin Settings. The password must not contain spaces. The default password is “456”.

Setting up Polycom Touch Control log management

It is possible to transfer the Polycom Touch Control logs to an external USB storage device.

To configure Polycom Touch Control log management:

1. Ensure that a USB device is connected to the USB port on the right side of the Polycom Touch Control device.

2. From the Home screen touch 🛠.
3. Under **Log Management**, select **Transfer Touch Control Logs to USB Device**.

Updating Polycom Touch Control Software

Software updates may be received for the Polycom Touch Control device from the online software server hosted by Polycom or from a USB storage device that you connect to the side of the Polycom Touch Control device.

To install Polycom Touch Control updates from the software server automatically:

1. From the Home screen, touch 🛠 and then **Updates.**
2. Ensure the correct server address is entered in the Server Address field. To use the Polycom server, enter Polycom. The field is not case sensitive.
3. Enable **Automatically Check for Software Updates**.
4. Specify the automatic update options:
   - **Start Time**: Touch **Hour**, **Minute** and **AM/PM** to specify the beginning of the time window within which the Polycom Touch Control device checks for updates.
   - Touch **Duration** to select the length of the time window within which the Polycom Touch Control device can check for updates.

     After the **Start Time** and **Duration** settings are configured, the Touch Control device calculates a random time within the defined update window at which to check for updates. It will then check for updates at this time on a daily basis as long as the **Start Time** and **Duration** values do not change. If the **Start Time** or **Duration** values change, a new random time within the new time window is calculated.
   - Touch **Action for Available Software Updates** and select whether to be notified of available status updates only or to download and install software when updates are available

To install Polycom Touch Control updates from the software server manually:

1. From the Home screen, touch 🛠 and then **Updates.**
2. Ensure the correct server address is entered in the Server Address field. To use the Polycom server, enter Polycom. The field is not case sensitive.
3. Touch **Check for Software Updates**.
4. Touch **Select All Updates** or touch only the updates that you want to install.
5. Touch **Download and Install Software Updates**.

To install Polycom Touch Control updates from a USB storage device:

1. On a computer, open Internet Explorer version 6.x, 7.x, or 8.x.
2. Go to support.polycom.com, and navigate to the page for the Polycom Video Codec system that you will use with the Polycom Touch Control.
3. Download the application and platform software update packages to your hard drive:
   - `polycom-venus-HDXCtrl-<version>.zip`
   - `polycom-venus-platform-<version>.zip`
4. Using a standard Windows zip utility, extract all contents from the distribution package or packages to the root directory of a USB storage device.

   When extracting multiple distribution packages to the USB drive, a pop up message might appear asking if you want to overwrite certain files that already exist. Select **Yes to All**.
5. Connect the USB device to the side of the Polycom Touch Control device.
6. From the Home screen, touch 🛠 and then **Updates.**
7. Touch **Check for Software Updates**.
8. Touch **Select All Updates** or touch only the updates that you want to install.

9. Touch **Download and Install Software Updates**.

Using the Polycom Touch Control with SoundStructure

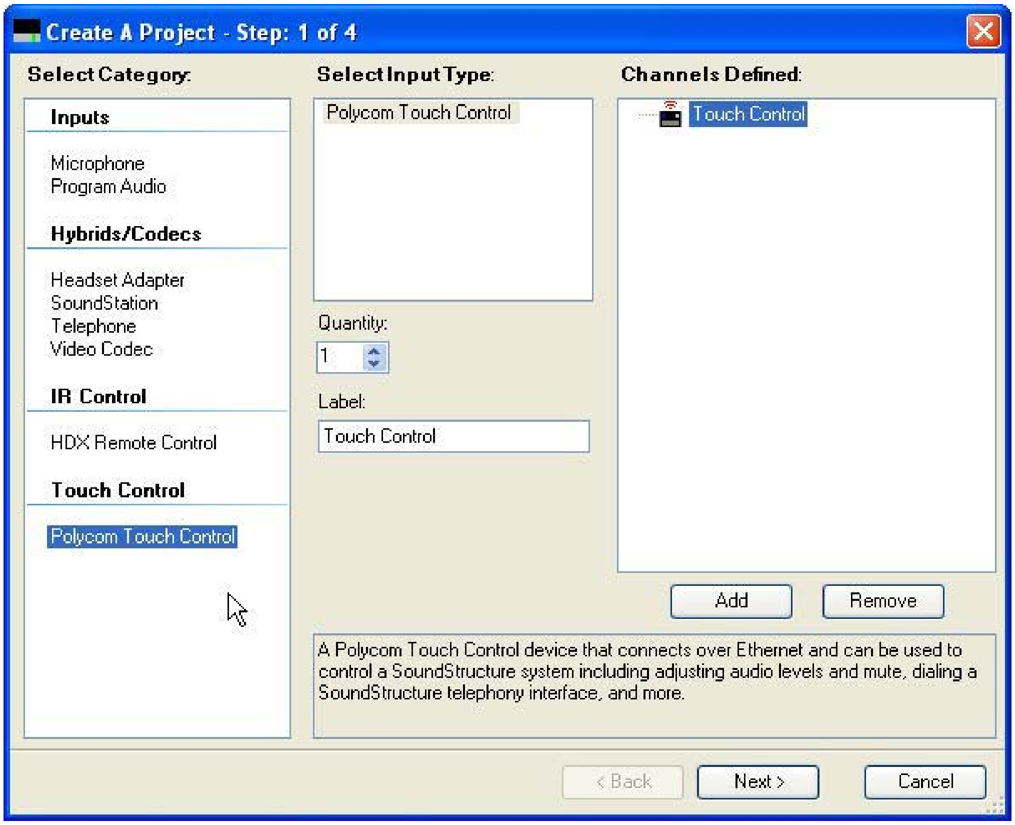
Designing a SoundStructure Project with the Polycom Touch Control

SoundStructure Studio version 1.5.0 or later includes support for creating a project that uses the Polycom Touch Control.

Adding a Touch Control to a SoundStructure project

SoundStructure Studio supports the Polycom Touch Control and allows one to be added to a project as shown in the following figure.

Adding Polycom Touch Control in SoundStructure Studio


Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Headset Adapter
SoundStation
Telephone
Video Codec
IR Control
HDX Remote Control
Touch Control
Polycom Touch Control
Select Input Type:
Polycom Touch Control
Channels Defined:
Touch Control
Add
Remove
< Back
Next >
Cancel

Quantity: 1

Label: Touch Control

A Polycom Touch Control device that connects over Ethernet and can be used to control a SoundStructure system including adjusting audio levels and mute, dialing a SoundStructure telephony interface, and more.

While up to four Polycom Touch Controls can be added into a project, only one Polycom Touch Control is necessary to create the required events. The additional three supported Polycom Touch Controls are planned for future use.

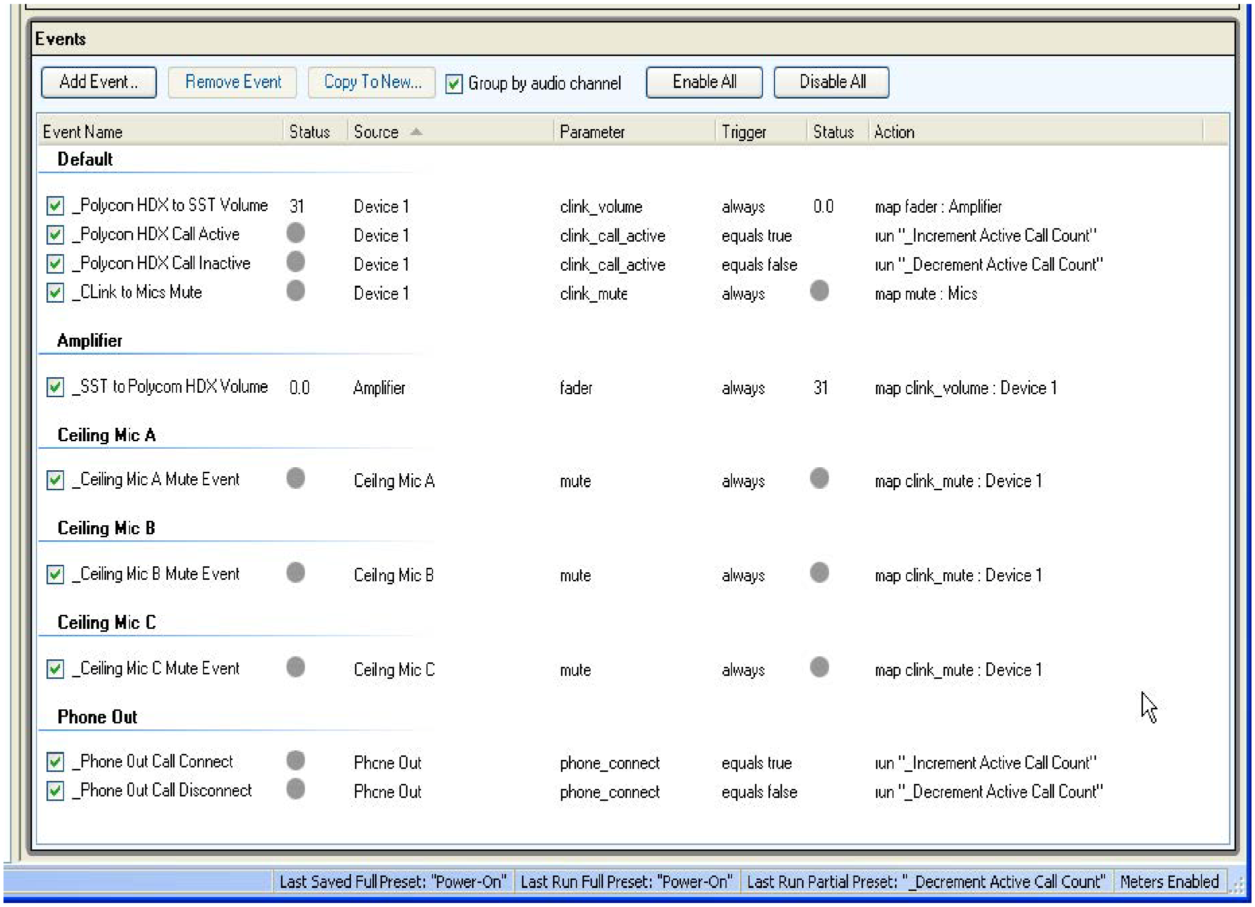
Polycom Touch Control causes Events to be created

SoundStructure Studio will automatically create the appropriate integration events when a Polycom Touch Control is added to the SoundStructure Studio project. The events that are created are the same events that would be created automatically if a Video Codec system were added to the SoundStructure Studio project.

In this system, a Polycom microphone channel and a SoundStructure PSTN telephony channel are also part of the project and there are events that were created to support those channels. See Connecting Over Conference Link2 for information about the events that are created for microphones and SoundStructure telephony channels.

The events that are created by SoundStructure Studio specifically for the Polycom Touch Control are shown in the following figure and described below. These events are sorted by their input virtual channel names.

. Polycom Touch Control Events



Event Name	Status	Source	Parameter	Trigger	Status	Action
Default						
☑ _Polycom HDX to SST Volume	31	Device 1	clink_volume	always	0.0	map fader : Amplifier
☑ _Polycom HDX Call Active	●	Device 1	clink_call_active	equals true		run "_Increment Active Call Count"
☑ _Polycom HDX Call Inactive	●	Device 1	clink_call_active	equals false		run "_Decrement Active Call Count"
☑ _CLink to Mics Mute	●	Device 1	clink_mute	always	●	map mute : Mics
Amplifier						
☑ _SST to Polycom HDX Volume	0.0	Amplifier	fader	always	31	map clink_volume : Device 1
Ceiling Mic A						
☑ _Ceiling Mic A Mute Event	●	Ceilng Mic A	mute	always	●	map clink_mute : Device 1
Ceiling Mic B						
☑ _Ceiling Mic B Mute Event	●	Ceilng Mic B	mute	always	●	map clink_mute : Device 1
Ceiling Mic C						
☑ _Ceiling Mic C Mute Event	●	Ceilng Mic C	mute	always	●	map clink_mute : Device 1
Phone Out						
☑ _Phone Out Call Connect	●	Phone Out	phone_connect	equals true		run "_Increment Active Call Count"
☑ _Phone Out Call Disconnect	●	Phone Out	phone_connect	equals false		run "_Decrement Active Call Count"

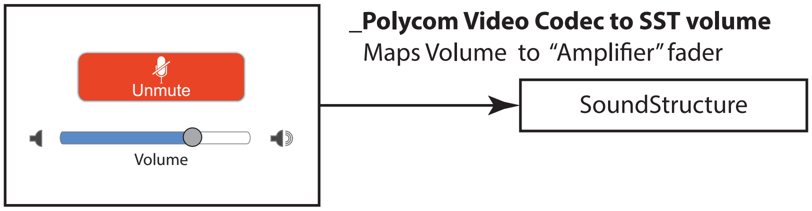
Last Saved Full Preset: "Power-On" | Last Run Full Preset: "Power-On" | Last Run Partial Preset: "_Decrement Active Call Count" | Meters Enabled

_Polycom Video Codec to SST Volume

The '_Polycom Video Codec to SST Volume' event maps the volume information from the Polycom Touch Control volume slider to the fader parameter of the channel called "Amplifier".

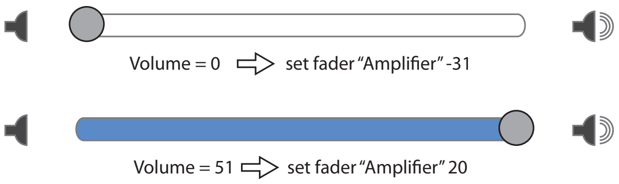
Adjusting the volume on the Polycom Touch Control adjusts the value of the fader parameter of the channel "Amplifier". This event may be customized to support different channels or different channel names. "Amplifier" is the name used by default.

Adjusting Polycom Touch Control Volume and Fader


Unmute
Volume
_Polycom Video Codec to SST volume
Maps Volume to "Amplifier" fader
SoundStructure

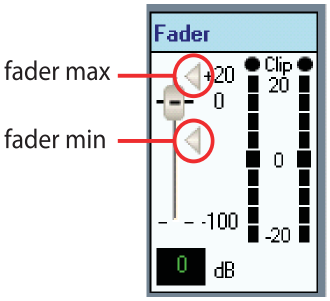
By default, SoundStructure Studio automatically sets the fader min of the “Amplifier” channel to -31 and the fader max to +20. This maps the volume range of the Polycom Touch Control slider from 0 to 51 to the fader range of -31dB to +20dB as shown in the following figure.

Polycom Touch Control Volume and Fader Controls


Volume = 0
set fader "Amplifier" -31
Volume = 51
set fader "Amplifier" 20

If a different fader min and max range is desired, new fader min and max values may be set by clicking and dragging the fader min and max controls on the Channels page within SoundStructure Studio and then saving the settings into the project.

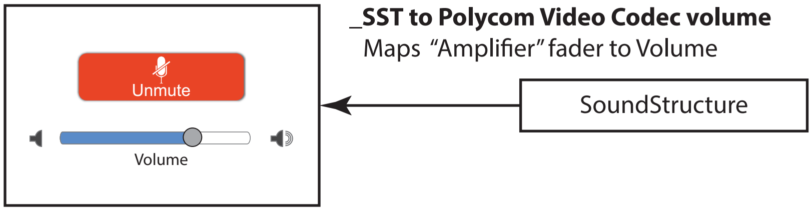
Fader Range


fader max
fader min
Fader
+20
0
-100
Clip
20
0
-20
0
dB

_SST to Polycom Video Codec Volume

The ‘_SST To Polycom Video Codec Volume’ event is the companion to the ‘_Polycom Codec to SST Volume’ event and maps the fader parameter of the "Amplifier" channel to the volume level on the Polycom Touch Control. Adjusting the volume of the fader of the "Amplifier" channel through some other control mechanism, such as through SoundStructure Studio or a Control System, adjusts the volume setting on the Polycom Touch Control. This event may be customized to support different channels. "Amplifier" is the name used by default.

_SST to Polycom Video Codec Amplifier Event


Unmute
Volume
_SST to Polycom Video Codec volume
Maps "Amplifier" fader to Volume
SoundStructure

_Clink to Mics Mute

The '_Clink to Mics Mute' event maps the mute status from the Polycom Touch Control to the mute of the virtual channel group "Mics". Muting or unmuting the system via the Polycom Touch Control maps that same mute state to the channel or group called "Mics". "Mics" is the name used by default.

_SST to Polycom Video Codec Mic Mute Event

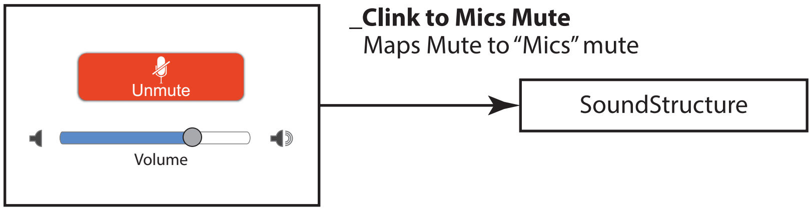
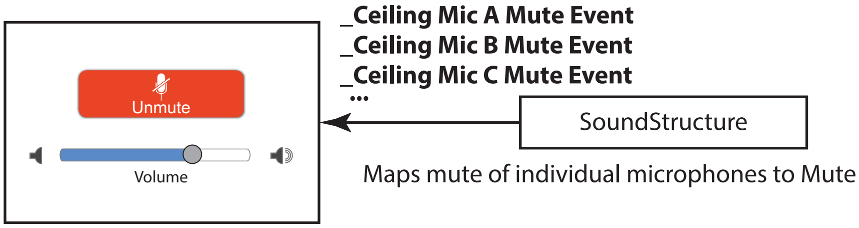

Unmute
Volume
_Clink to Mics Mute
Maps Mute to "Mics" mute
SoundStructure

_Table Mic A Mute Event, _Table Mic B Mute Event, _Table Mic C Mute Event, …

These events map the individual microphone mute status from push to talk microphones or Codec microphones to the clink_mute parameter that is used to set the mute state of the system. Muting any of the individual microphones will set clink_mute to 1 which will then mute all microphones due to the _Clink to Mics Mute event which then causes the Polycom Touch Control to show that the system is muted.

_Ceiling Mic Mute Event


Unmute
Volume
_Ceiling Mic A Mute Event
_Ceiling Mic B Mute Event
_Ceiling Mic C Mute Event
...
SoundStructure
Maps mute of individual microphones to Mute

Using Multiple SoundStructure Telephony Interfaces

Multiple SoundStructure telephony interfaces are supported with the Polycom Touch Control as follows:

- When a SoundStructure VoIP Interface is used, only the first registered VoIP line can be dialed with a Polycom Touch Control. Additional lines are ignored.
  - When multiple telephone calls are active in the system, the SoundStructure telephony lines are used sequentially, as sorted alphabetically by the telephone line virtual channel names.
  - The first telephone line will be used when the user dials the first phone call. Subsequent phone lines will be used if the user adds another call to the existing call by pressing + Add Call.
  - Once all the SoundStructure telephone interfaces are in active calls, no more telephone calls may be added to the system. If the user tries to add another call, a message indicating that the “meeting is full” will be displayed. This means that no more calls may be added to the system.
  - All calls may be hung up at the same time or alternatively individual lines, displayed by virtual channel name, may be hung up.

See the User’s Guide for SoundStructure Systems and the Polycom Touch Control for additional information about using the Polycom Touch Control.

Note: Multiple Phone Lines Not Controlled Separately with Polycom Touch Control

While multiple SoundStructure telephony interfaces are supported, multiple calls are assumed to be used in the same room. Multiple independent telephone are used sequentially when multiple callers are brought into the conference. Multiple phone lines cannot be controlled independently (for example combined and divided room applications) with the Polycom Touch Control.

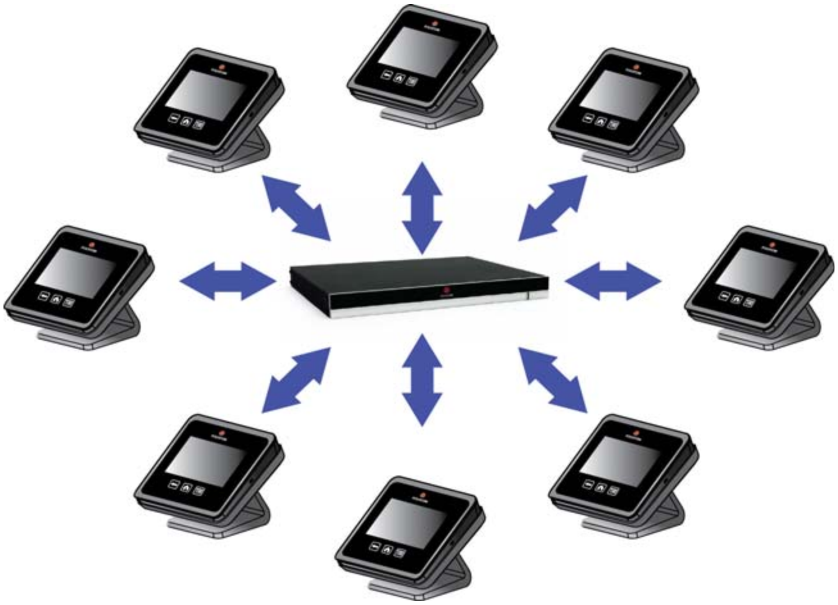
When

Using Multiple Polycom Touch Controls with SoundStructure

After a Polycom Touch Control has been designed in the system and the events are created, multiple Polycom Touch Controls may be paired with the SoundStructure system and used to control the SoundStructure system. An application where multiple Polycom Touch Controllers would be useful would be in a large room where multiple locations in the room would like to have dialing, volume, and mute control.

All Polycom Touch Controls paired with a SoundStructure system operate synchronously and control the same aspects of SoundStructure system. For example, multiple Polycom Touch Controls may be used to mute and adjust volume in the system and all Polycom Touch Controls will show the same mute and volume status.

Multiple Polycom Touch Controls Paired with a SoundStructure System



Validating Polycom Touch Control and SoundStructure integration

If the Polycom Touch Control and SoundStructure are paired and configured properly then volume changes on the Polycom Touch Control will be heard directly in the local room and microphones will be muted when the mute button is pressed.

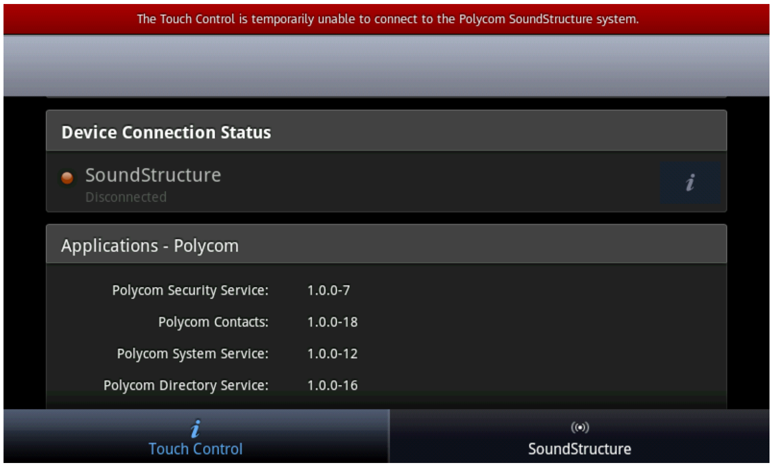
If the Polycom Touch Control is not properly controlling the SoundStructure system, there are several steps to follow.

Paired with the proper SoundStructure system?

To determine if the Polycom Touch Control is paired properly with the SoundStructure system:

1. Verify that the systems are paired by pressing ℹ.
2. Scroll down to **Device Connection Status** and press the ***i*** button next to the SoundStructure name. If there is no system listed as connected, then the Polycom Touch Control is not paired with a SoundStructure system. To pair the Polycom Touch Control with a SoundStructure system, touch the View Pairing Settings and follow the pairing instructions described earlier in this chapter. The Admin credentials login prompt may appear after pressing View Pairing Settings.

If the SoundStructure system is listed but shown as Disconnected as shown in the following figure, then the system was successfully paired, but the SoundStructure system is no longer accessible via the network. In this case there will also be a banner on the top of the screen that indicates that the connection has been temporarily lost. To resolve this situation check the network connections on both the Polycom Touch Control and the SoundStructure system.



The Touch Control is temporarily unable to connect to the Polycom SoundStructure system.

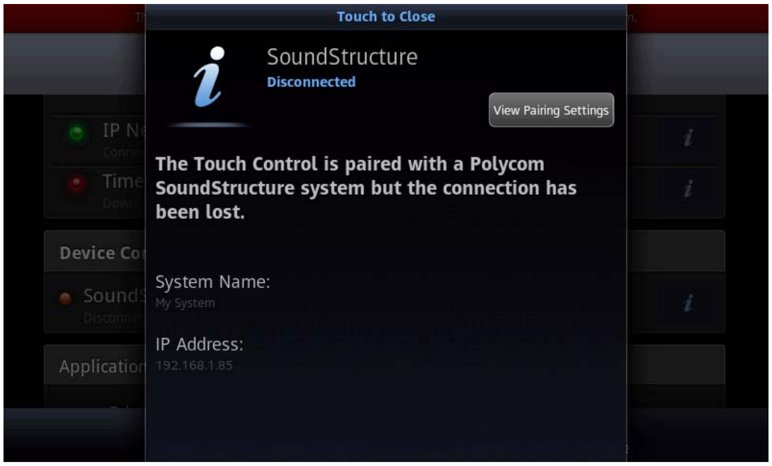
Device Connection Status

SoundStructure
Disconnected

Applications - Polycom

Polycom Security Service:	1.0.0-7
Polycom Contacts:	1.0.0-18
Polycom System Service:	1.0.0-12
Polycom Directory Service:	1.0.0-16

Touching the ***i*** button shows that the Polycom Touch Control is paired with a system but the connection has been lost as shown in the following figure.



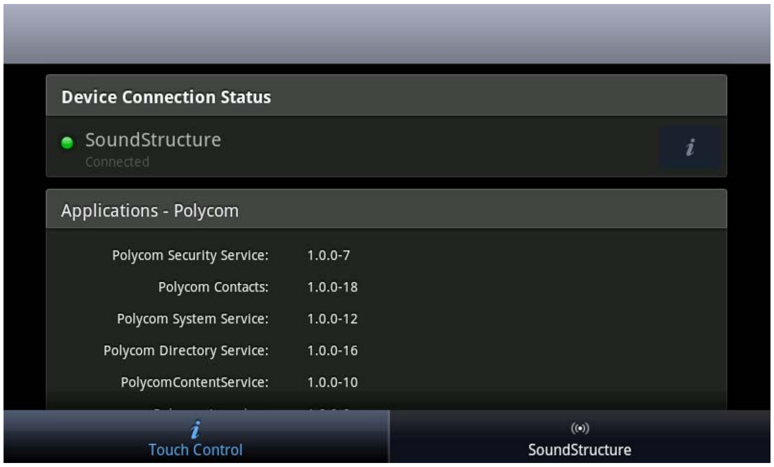
SoundStructure
Disconnected

The Touch Control is paired with a Polycom SoundStructure system but the connection has been lost.

System Name:
My System

IP Address:
192.168.1.85

3 If the system is properly paired and the Polycom Touch Control and the SoundStructure system are communicating then the display will appear as shown in the following figure.


Device Connection Status

SoundStructure
Connected

Applications - Polycom

Polycom Security Service:	1.0.0-7
Polycom Contacts:	1.0.0-18
Polycom System Service:	1.0.0-12
Polycom Directory Service:	1.0.0-16
PolycomContentService:	1.0.0-10

Are the SoundStructure system’s events defined properly?

If the Polycom Touch Control is paired with the proper SoundStructure system, the next step to check is whether the events are properly defined within the SoundStructure system.

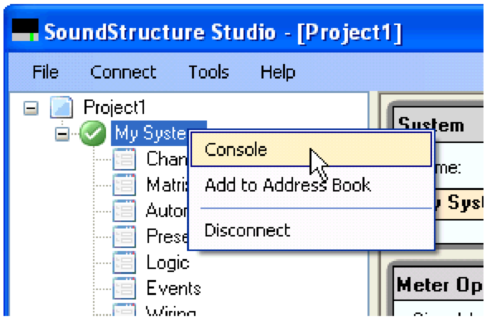
1 Connect to the SoundStructure system with SoundStructure Studio and check that the events are shown as described previously.

2 Check that the events are using the proper channel names for system that is being controlled.

Each event should have a valid Action that specifies the virtual channel name that will be affected. For example, the _Polycom Codec to SST Volume event should specify the source of the event (clink_volume on Device 1), the trigger (always), and the action map the value to the fader of the channel “Amplifier”. If the “Amplifier” channel name is missing or does not match the desired channel that should be adjusted when the volume is changed, then edit the event by double clicking on the event. See Using Events, Logic, and IR for more information on creating and editing events.

If the events are defined properly, the command acknowledgments are seen in the SoundStructure console. To open the console, right click on the project name in SoundStructure Studio as shown in the following figure.

Opening the Console in SoundStructure Studio


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
My Syste
Chan
Matri
Auto
Prese
Logic
Events
Wiring
Console
Add to Address Book
Disconnect
System
me:
Syst
Meter Op

To see SoundStructure command acknowledgments, press the mute button on the Polycom Touch Control. When the system is muted, the console shows text such as:

```
val mute "Table Mic 1 A" 1
val mute "Table Mic 1 B" 1
val mute "Table Mic 1 C" 1
val mute "Table Mic 2 A" 1
val mute "Table Mic 2 B" 1
val mute "Table Mic 2 C" 1
val mute "Mics" 1
val clink_mute 1 1
```

which confirms that the SoundStructure system acted on the mute and responded with acknowledgments that the system is now muted.

If these acknowledgments are not seen, verify that the events are defined properly and that the console for the appropriate SoundStructure system has been opened.

Did the telephony virtual channel names change?

If the virtual channel name for a SoundStructure telephony interface is changed after the Polycom Touch Control has been paired with a SoundStructure system, it will neither be possible to dial that telephony interface nor receive incoming calls on that telephony interface because the Polycom Touch Control will not be using the proper channel name within the API commands sent to the SoundStructure system.

If any of the telephony virtual channel names have been changed after the system has been paired, it will be necessary to either

- reboot the Polycom Touch Control, or
- re-pair the Polycom Touch Control with the SoundStructure system

to ensure the Polycom Touch Control is using the correct telephony virtual channel names.

Integrating SoundStructure with SoundStructure VoIP Interface

In this chapter you will learn how to create a SoundStructure design with the SoundStructure VoIP Interface and how to integrate that SoundStructure design into a SIP environment.

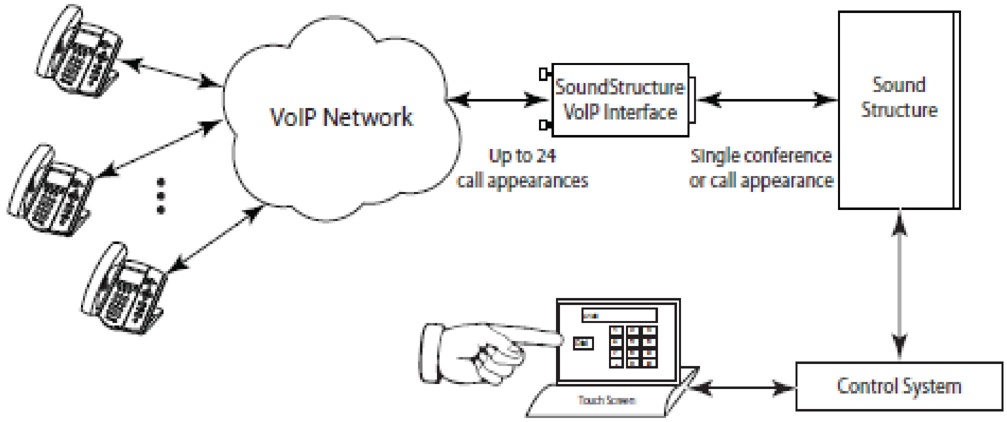
This chapter begins with an introduction to the SoundStructure VoIP Interface, followed by detailed information on setting up the SoundStructure VoIP Interface.

Introduction

The SoundStructure VoIP Interface is a plug-in card for SoundStructure systems that is a high-quality SIP-based end-point. This card has been designed to inter-operate with other compatible equipment including application servers, media servers, Internet-working gateways, voice bridges, and other end-points. The SoundStructure VoIP Interface uses the industry-leading Polycom UC Software that is the foundation of the SoundPoint IP and SoundStation IP phones.

The SoundStructure system can be controlled by a third-party remote control system. The control system will send API commands to the SoundStructure system to cause the SoundStructure system to take the SoundStructure VoIP Interface offhook, dial digits, put calls on hold, resume calls, and more. A typical call and control scenario with the user interacting with the touch screen is shown next.

SoundStructure VoIP Interface Call and Control Scenario


VoIP Network
SoundStructure VoIP Interface
Up to 24 call appearances
Single conference or call appearance
Sound Structure
Touch Screen
Control System

This chapter describes in detail how to create a fully functional SoundStructure VoIP Interface-based system. From a high-level perspective, there are only a few things that need to be done once you have the proper firmware and version of SoundStructure Studio:

1. Create a SoundStructure Studio project with the SoundStructure VoIP Interface by creating a new project or upgrading an existing project.
2. Install the SoundStructure VoIP Interface plug-in card into the SoundStructure system.
3. Give the MAC address of the SoundStructure VoIP Interface to the IT/Phone team and receive the SIP server IP address and VoIP phone line registration information.
4. Enter the SoundStructure VoIP Interface the SIP server IP address and line registration information from the Web UI of the SoundStructure VoIP Interface
5. Make a test call to confirm the line is properly registered to the SIP server.

The rest of the chapter explains these step in detail.

How to Read This Chapter

Use the following table to decide which section will help you.

Chapter Topics

If you...	Then see...
Have an existing system with a TEL1 or TEL2 and want to upgrade to a SoundStructure VoIP Interface	Upgrading an Existing TEL1/TEL2 Project to the SoundStructure VoIP Interface
Want to create a new project with a SoundStructure VoIP Interface	Creating a New Project with the SoundStructure VoIP interface
Want to dial a call or change the phone settings using SoundStructure Studio	Using the SoundStructure VoIP Interface with SoundStructure Studio
Need to configure a SoundStructure VoIP Interface to work with a SIP call platform.	Configuring the SoundStructure VoIP Interface
Want to validate your installation of the SoundStructure VoIP Interface	Validating a SoundStructure VoIP Interface Installation
Update the software on your SoundStructure VoIP Interface	Updating Software on the SoundStructure VoIP Interface
Want to see the new API commands and examples	Understanding SoundStructure VoIP Interface API Commands
Want to learn more about administration of VoIP systems	Polycom UC Software Administrators Guide 4.0.1

SoundStructure VoIP Interface Overview

The SoundStructure VoIP interface is a plug-in card designed for use with the rear panel slot available on all SoundStructure devices. One SoundStructure VoIP Interface may be used per SoundStructure device. If there is a system of multiple SoundStructure devices, each SoundStructure device can have a SoundStructure VoIP Interface installed.

SoundStructure VoIP Interface Plug-In Card

POLYCOM
Status
VoIP LAN

The SoundStructure VoIP Interface has the following capabilities:

- 12-line support where any single call or a conference of calls can be sent to the SoundStructure device.
- 24 call appearances where any call appearance or conference of call appearances can be sent to the SoundStructure device.
- Supports the following narrowband audio codecs:
  - G.711 m/A,
  - G.729A (Annex B),
- Supports the following wide-band audio codecs:
  - G.722 (7 kHz),
  - G.722.1 (7 kHz),
  - G.722.1C (14 kHz)
- Supports SIP signaling and many SIP call features.
- Compatible with many SIP call management platforms.
- Supports API command set for dialing, on hook, offhook, call hold, call resume, call transfer, call blind transfer, call forward, call hangup, call conference, call split, call join, and do not disturb.
- Easy to upgrade from an existing SoundStructure TEL1 or TEL2 system.

SoundStructure TEL1/TEL2 to SoundStructure VoIP Interface Considerations

To better understand how the SoundStructure VoIP Interface relates to a TEL1 and TEL2 interface card, consider the following points:

- One SoundStructure VoIP Interface may be used per SoundStructure device, just like a TEL1 or a TEL2 interface card.
- A SoundStructure VoIP Interface works like the TEL1 interface and supports one independent telephone call to the SoundStructure system. Just like the TEL1, one independent call is sent to the SoundStructure audio matrix. However, this one independent call can be the mix of two remote participants. See the command examples at the end of this chapter for more information on dialing two remote participants.
- Support for 12 “lines” means that 12 different extensions can be defined for this card to provide different dial-in numbers for remote participants if necessary.

- Supporting 24 call appearances means that 24 remote participants can be dialed into the SoundStructure VoIP Interface and one remote participant or a conference of two remote participants can be active and that mix of remote participants sent to the SoundStructure system. All other call appearances that are not part of the call sent to the SoundStructure VoIP Interface would be on hold.
- The SoundStructure VoIP Interface supports calling more than one remote participant in the same conference. This means that, if you are using a TEL2 card to support dialing two remote parties into the same call, you can also do this with the SoundStructure VoIP Interface. See the example of calling two remote parties: Dialing Two Calls on the Same Line.
- Two independent calls are not supported on the SoundStructure VoIP Interface. In other words, if you are using a TEL2 card to support two independent telephone calls, such as for a split/combine room operation, you will need two SoundStructure VoIP Interfaces.

SoundStructure System Requirements

To use the SoundStructure VoIP Interface, the following versions of software are required. To get the latest software versions, please visit Polycom Support and download the versions of software for your SoundStructure system.

SoundStructure Firmware version 1.5

This firmware version is fully compatible with configuration files created with earlier versions of SoundStructure Studio.

SoundStructure Studio version 1.7

This studio version is fully compatible with configuration files created with earlier versions of SoundStructure Studio. Once a configuration file has been created with version 1.7.0 it may not be backwardly-compatible with earlier versions.

SoundStructure VoIP Interface Firmware version 4.0.1

Support for the SoundStructure VoIP Interface requires version 4.0.1 or later of the Polycom UC Software for the SoundStructure VoIP Interface. By default, the SoundStructure VoIP Interface product ships with version 4.0.1 or later installed.

Upgrading a Project to the SoundStructure VoIP Interface

An existing SoundStructure project with PSTN interface may be upgraded to the SoundStructure VoIP Interface with the Upgrade Plug-in Card tool. This new tool has been designed to upgrade a SoundStructure project from one set of telephony plug-in cards to a different set of plug-in cards. This section shows you how to use the Upgrade Plug-in Card Tool with an example that upgrades a SoundStructure project from a SoundStructure TEL1 card to the SoundStructure VoIP Interface.

For a summary of the all the steps required to update an install from a SoundStructure TEL1 to a SoundStructure VoIP Interface, see the SoundStructure VoIP Interface Quick Upgrade Guide on Polycom Support.

Upgrading an Existing TEL1/TEL2 Project to the SoundStructure VoIP Interface

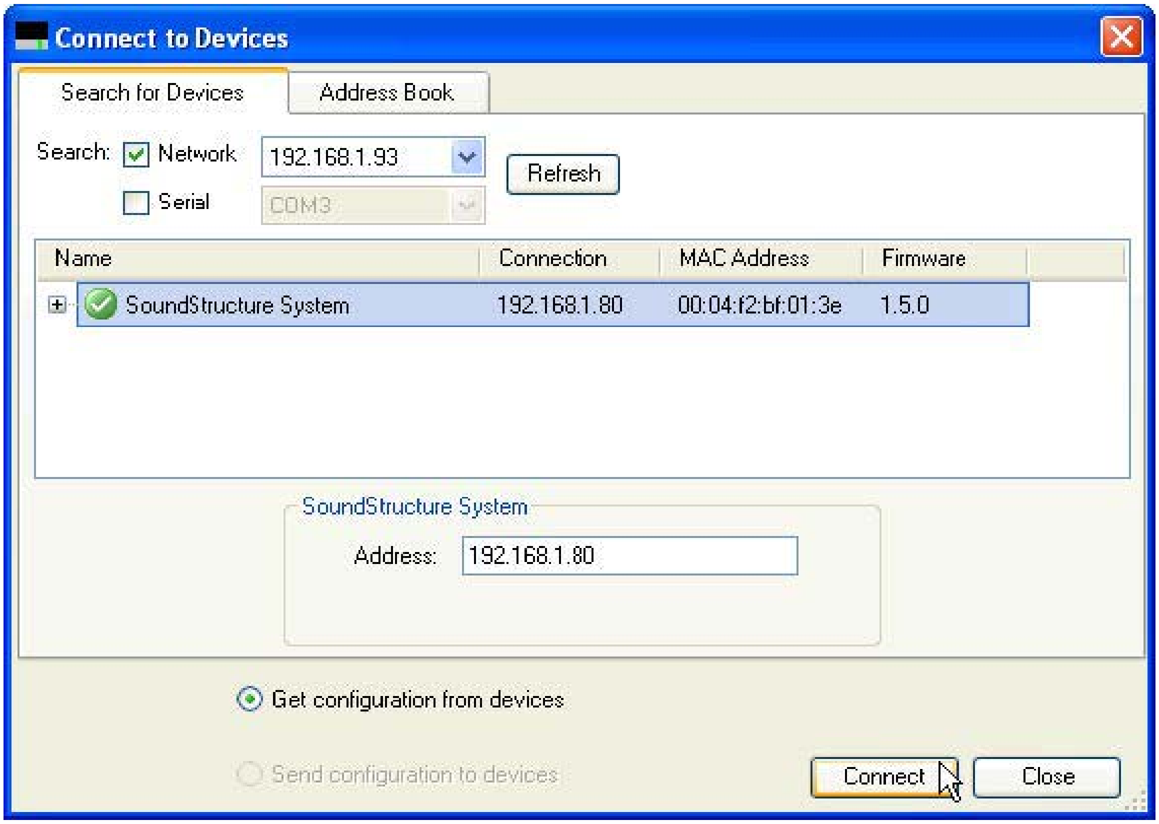
Polycom recommends upgrading telephony cards in a SoundStructure system by working offline with the project file and running the Upgrade Plugin Card Tool with the offline project.

Starting with an Online System

While SoundStructure Studio supports upgrading an online system, Polycom recommends that you get the configuration file from your system and then disconnect from the system and continue to work offline to reduce the number of steps and simplify the upgrade process.

To get the configuration file, select the **Connect** menu option in SoundStructure Studio and select the **Search for Devices** option to find the desired system from the list of discovered systems. If your system is not discovered, enter the IP address of the system or choose a system from the address book. Once the system is identified, select the **Get configuration from devices** option and select **Connect** as shown in the following figure.

Connecting to SoundStructure Devices



Search: ☑ Network 192.168.1.93

☐ Serial COM3

Name	Connection	MAC Address	Firmware
SoundStructure System	192.168.1.80	00:04:f2:bf:01:3e	1.5.0

SoundStructure System

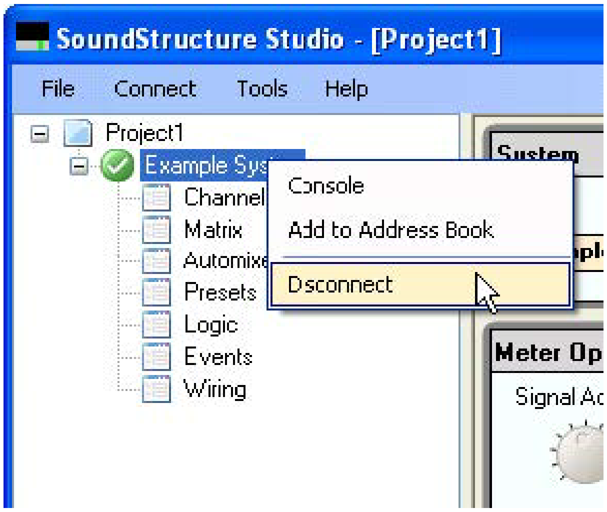
Address: 192.168.1.80

● Get configuration from devices

○ Send configuration to devices

After you have the retrieved the configuration file, disconnect from the online system by right-clicking on the project name and selecting **Disconnect** from the popup menu as shown next.

Disconnecting a SoundStructure Device

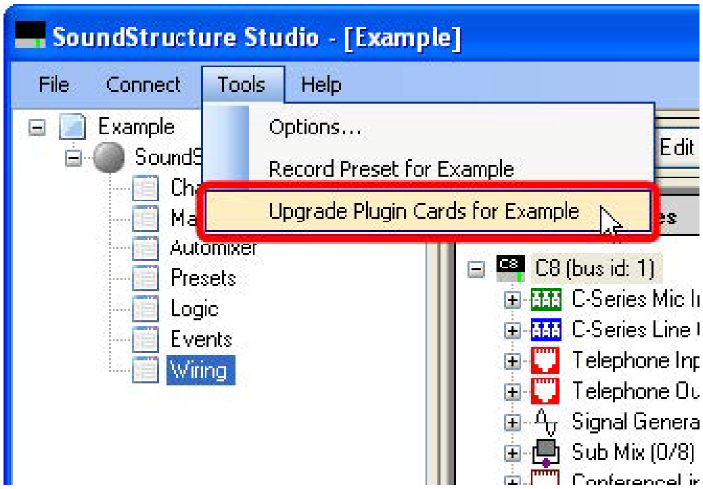

SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
Example Sys
Channel
Matrix
Automixe
Presets
Logic
Events
Wiring
Console
Add to Address Book
Disconnect
System
Meter Op
Signal Ac

Starting with an Offline System

Once you have an existing SoundStructure project, use the Upgrade Plug-in Card Tool to change plug-in cards. The Upgrade Plug-in Card Tool automatically preserves the telephony channel definitions and substitutes the new interfaces that you have selected, leaving presets and event definitions unaffected by the plug-in card conversion.

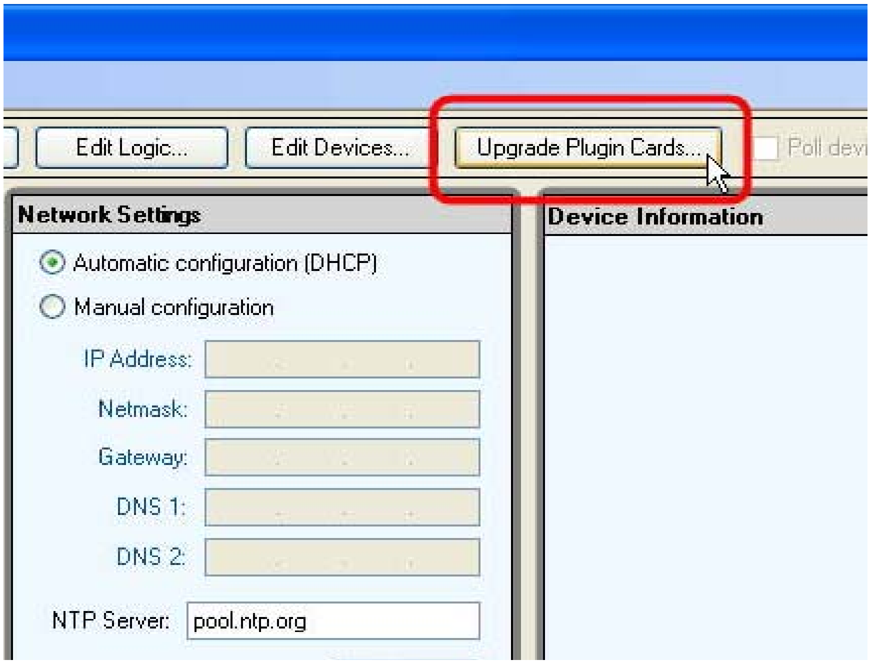
To change the telephony interfaces in an offline project, select **Upgrade Plug-in Card for Example** from the **Tools** menu as shown next.

Upgrade Plugin Card Example


SoundStructure Studio - [Example]
File
Connect
Tools
Help
Example
SoundS
Ch
Ma
Automixer
Presets
Logic
Events
Wiring
Options...
Record Preset for Example
Upgrade Plugin Cards for Example
Edit
C8 (bus id: 1)
C-Series Mic I
C-Series Line I
Telephone Inp
Telephone Ou
Signal Genera
Sub Mix (0/8)
Conference Lin

Or click **Upgrade Plug-in Card** from the **Wiring** page as shown next.

Upgrade Plugin Card


Edit Logic...
Edit Devices...
Upgrade Plugin Cards...
Poll devi
Device Information

Network Settings

● Automatic configuration (DHCP)

○ Manual configuration

IP Address:

Netmask:

Gateway:

DNS 1:

DNS 2:

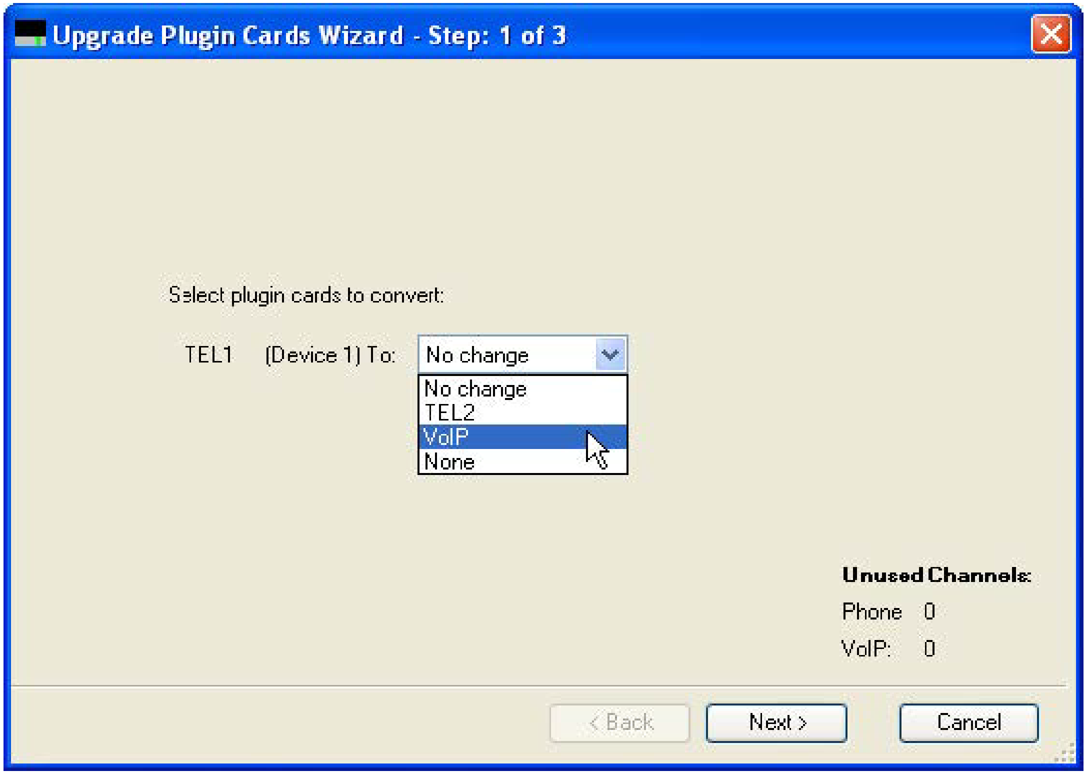
NTP Server: pool.ntp.org

Next, the steps required to change your plug-in cards are presented.

1. Select the plug-in cards to change. For each plug-in card in the SoundStructure system, you have the following options:
   - Leave the plug-in card as it is,
   - Change the plug-in card to a different card, or
   - Remove the plug-in card from the system.

Consider the example, shown below, where there is a single SoundStructure device and one plug-in card displayed. When there are multiple SoundStructure devices in the system, there may be a plug-in card entry for each device.

Upgrade Plugin Example



Upgrade Plugin Cards Wizard - Step: 1 of 3

Select plugin cards to convert:

TEL1 (Device 1) To: No change

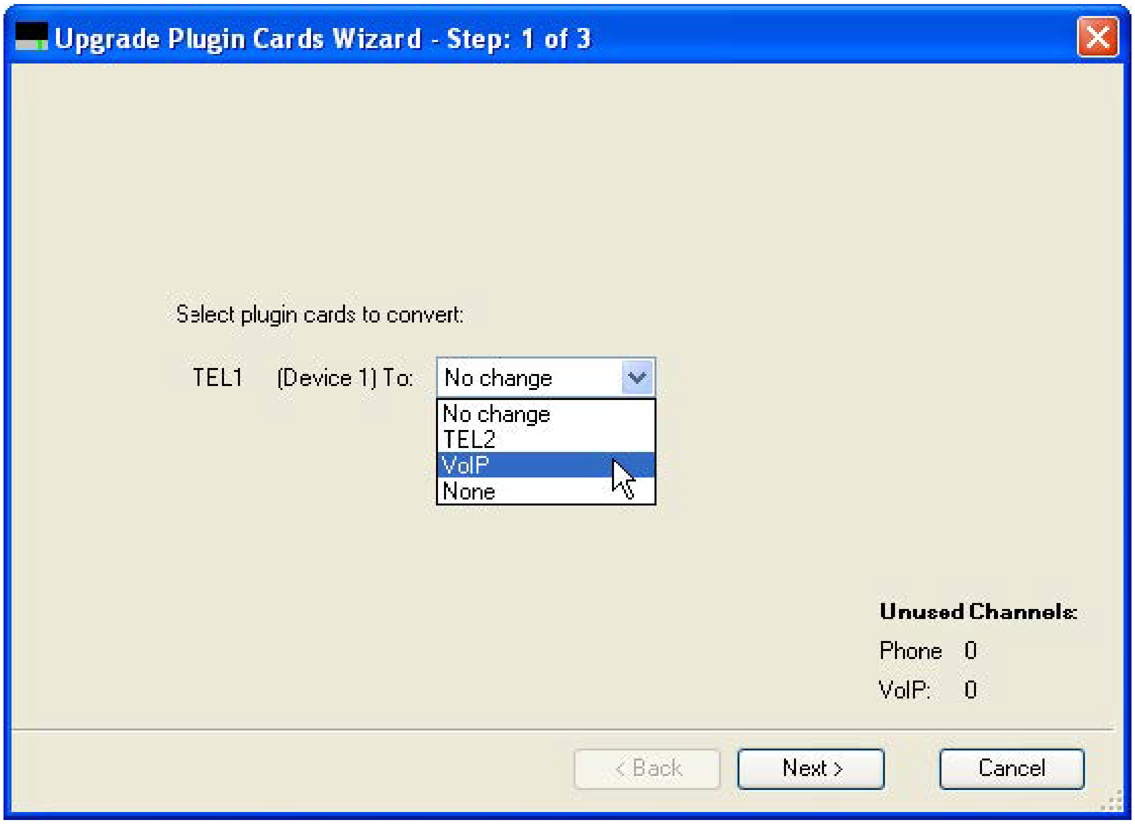
Unused Channels:

Phone 0

VoIP: 0

In this example, select the desired card to convert to a different card and click **Next**. In this example, the TEL1 plug-in card will be changed to a VoIP plug-in card.

Selecting Desired Plugin Example



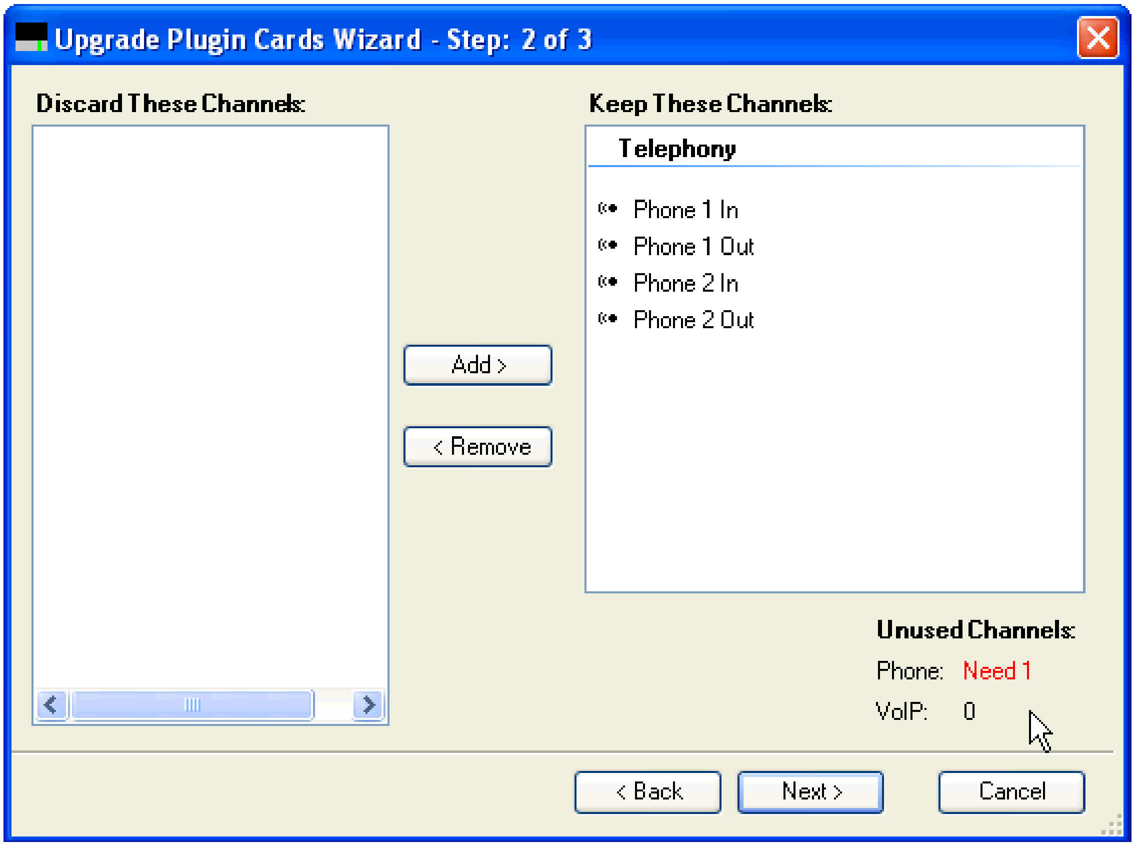
Upgrade Plugin Cards Wizard - Step: 1 of 3
Select plugin cards to convert:
TEL1 (Device 1) To:
No change
No change
TEL2
VoIP
None
Unused Channels:
Phone 0
VoIP: 0
< Back
Next >
Cancel

2 Remove Telephony Channels. If the number of telephony channels (the sum of the number of telephony channels on all interfaces) available in the converted project is less than the number of telephony channels defined in the project (the number of telephony virtual channel definitions in the project), extra telephony channel definitions must be removed from the project.

Consider the example of replacing a TEL2 that has two telephony channels defined in the source project with a TEL1 telephony card that supports only one telephony channel. In this example, as shown below, you must remove one telephony channel from the project before being able to continue with the project conversion. In this case the ‘Unused Channels’ field in the lower right-hand corner of the Upgrade Plug-in

Cards Tool window indicates it is necessary to remove one telephony channel from the system based on the number of telephony channels available.

Removing Telephony Channels

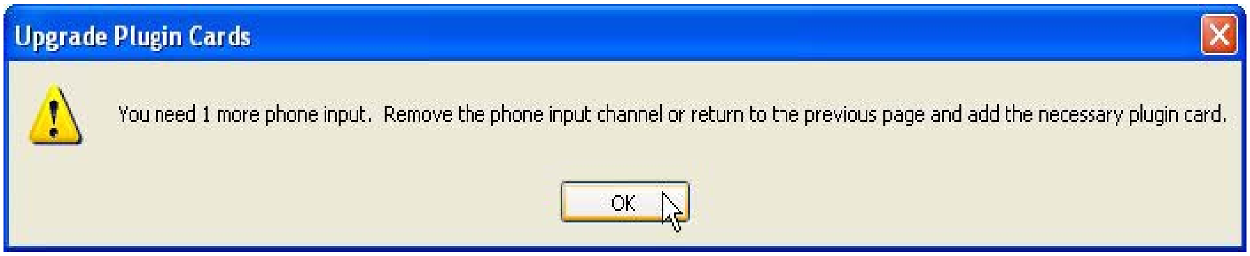

Upgrade Plugin Cards Wizard - Step: 2 of 3
Discard These Channels:
Keep These Channels:
Add >
< Remove
Unused Channels:
< Back
Next >
Cancel

Telephony
Phone 1 In
Phone 1 Out
Phone 2 In
Phone 2 Out

Phone: Need 1
VoIP: 0

If you click **Next** and there are not enough telephony resources available, the following alert is displayed.

Telephony Resources Alert


Upgrade Plugin Cards
OK

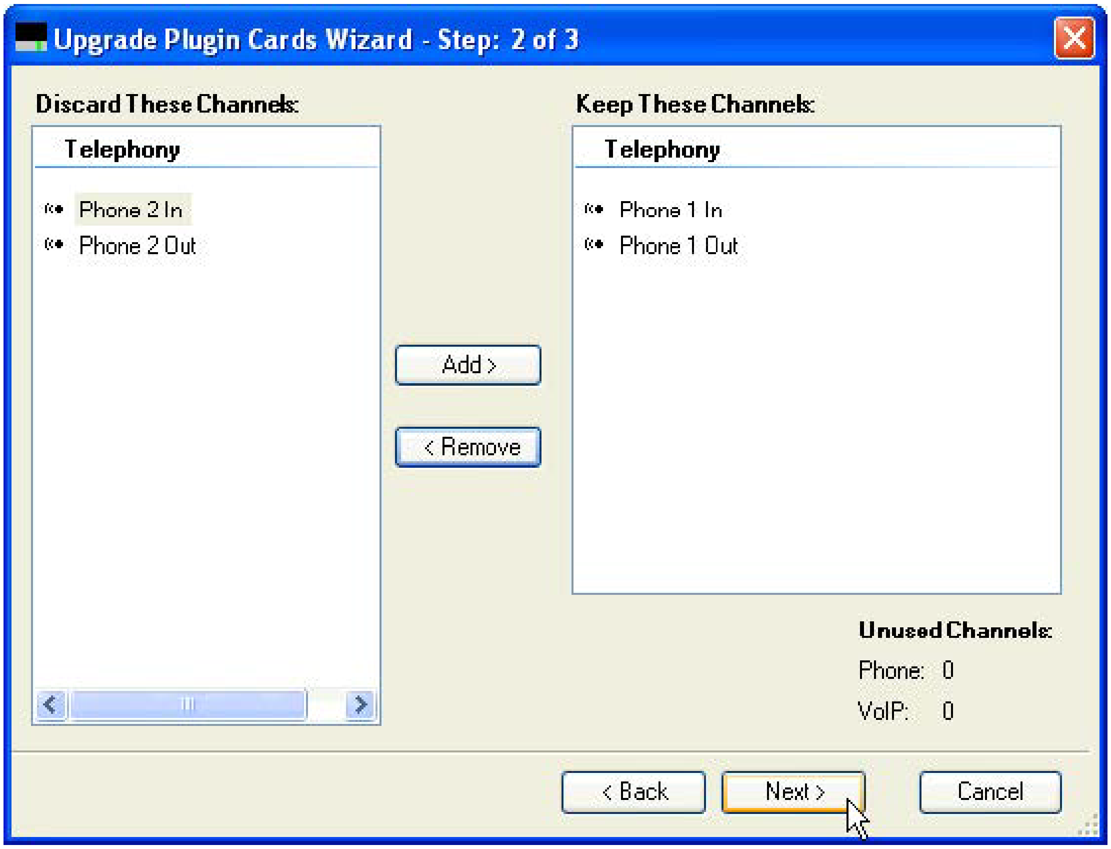
You need 1 more phone input. Remove the phone input channel or return to the previous page and add the necessary plugin card.

To complete the upgrade process in this situation, you need to either:

- Remove the telephony channel definition that will not be in the upgraded project after the transition from the TEL2 to the TEL1. Remove the channel by selecting it and clicking **Remove**, or

- ➢ Click **Back** and undo removing the TEL2 plug-in card from the system.

To remove the telephony channel, select either the Phone In or Phone Out channel and click **Remove**. Selecting either the input or output channel will remove the entire telephony channel. Deleting the channel will update the number of Unused Channels, and, if no further channels are needed, you may click **Next** to continue.


Upgrade Plugin Cards Wizard - Step: 2 of 3
Add >
< Remove
< Back
Next >
Cancel

Discard These Channels:

Telephony
- Phone 2 In
- Phone 2 Out

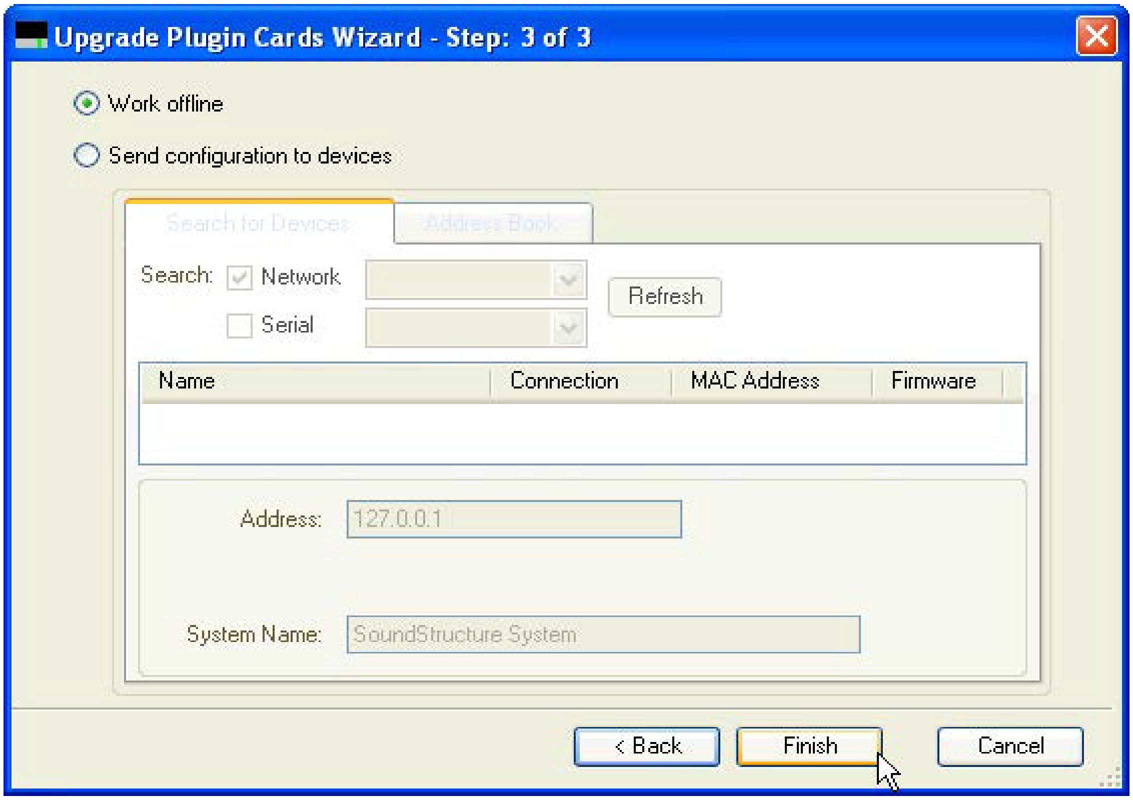
Keep These Channels:

Telephony
- Phone 1 In
- Phone 1 Out

Unused Channels:

Phone: 0
VoIP: 0

3 Decide to continue to work offline or send the project to an online SoundStructure system. Polycom recommends you continue to work offline.


Upgrade Plugin Cards Wizard - Step: 3 of 3
Search for Devices
Address Book
Refresh
Name
Connection
MAC Address
Firmware
< Back
Finish
Cancel

● Work offline

○ Send configuration to devices

Search: ☑ Network

☐ Serial

Address: 127.0.0.1

System Name: SoundStructure System

4 Click **Finish** to create the updated project with the new plug-in cards that you selected in the process. The project file may now be customized further if needed before sending to the SoundStructure system.

At this point, you have created an offline project that includes the SoundStructure VoIP Interface. Configuration of the SoundStructure VoIP Interface settings will be described in Configuring the SoundStructure VoIP Interface.

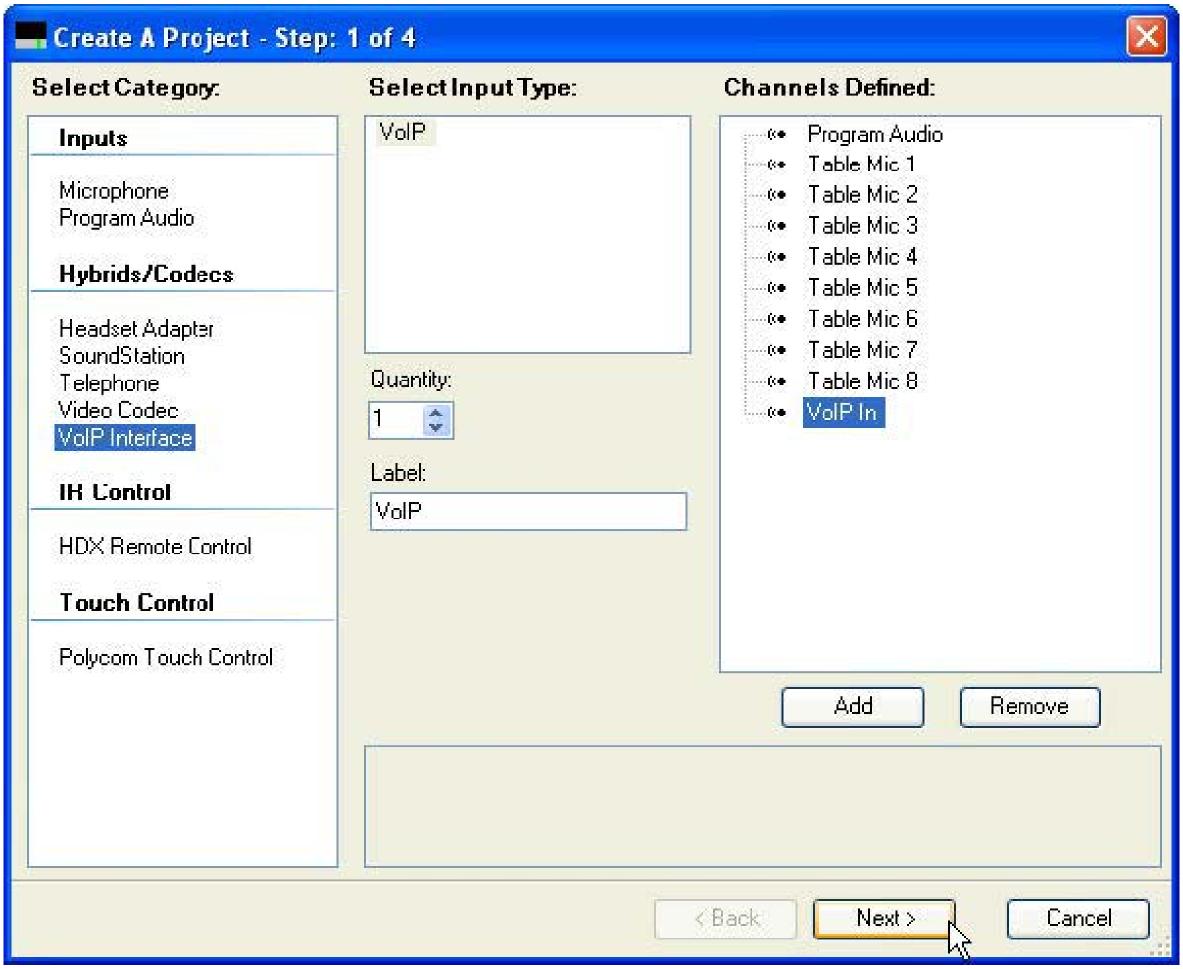
Creating a New Project with the SoundStructure VoIP interface

While the previous section focuses on converting an existing project, this section focuses on creating a new project with the SoundStructure VoIP Interface.

To create a new project, follow these steps.

1 Launch SoundStructure Studio, select the File menu, and then select **New Project**.

2 In *Create a Project - Step 1*, select the inputs to be used in this project and after each selection click **Add**. The following figure shows an example system with 8 analog microphones (Table Mics 1 through 8), a SoundStructure VoIP interface (VoIP In), and mono program audio input (Program Audio).



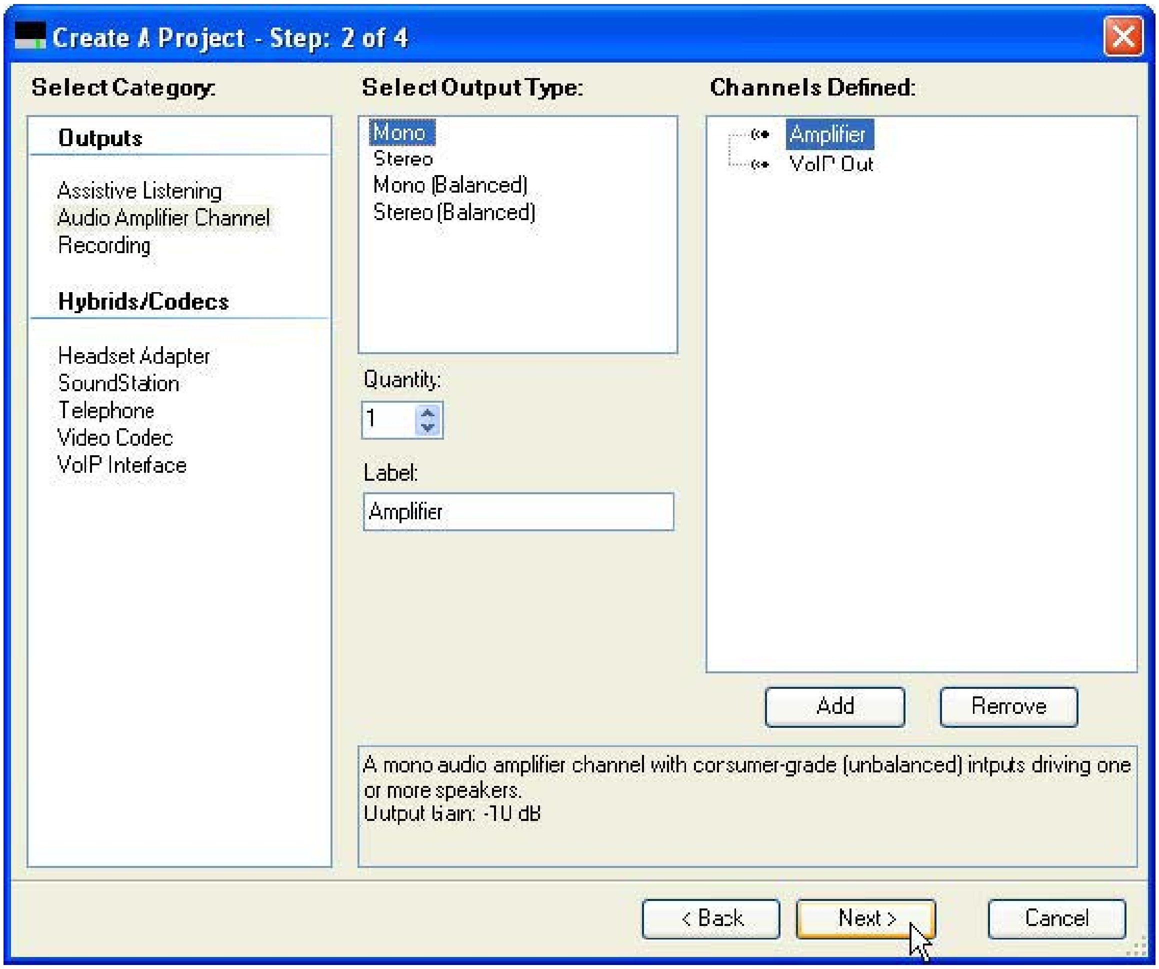
Quantity: 1

Label: VoIP

Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Headset Adapter
SoundStation
Telephone
Video Codec
VoIP Interface
IR Control
HDX Remote Control
Touch Control
Polycom Touch Control
Select Input Type:
VoIP
Channels Defined:
Program Audio
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
VoIP In
Add
Remove
< Back
Next >
Cancel

3 Click **Next** to select the outputs from the system.

4 In *Create A Project - Step 2,* select a mono audio amplifier and click **Add**. The resulting system will look like the following figure. Notice the VoIP Out channel was automatically added when the SoundStructure VoIP Interface was selected.


Create A Project - Step: 2 of 4
Select Category:
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Headset Adapter
SoundStation
Telephone
Video Codec
VoIP Interface
Select Output Type:
Mono
Stereo
Mono (Balanced)
Stereo (Balanced)
Channels Defined:
Amplifier
VoIP Out
Add
Remove
< Back
Next >
Cancel

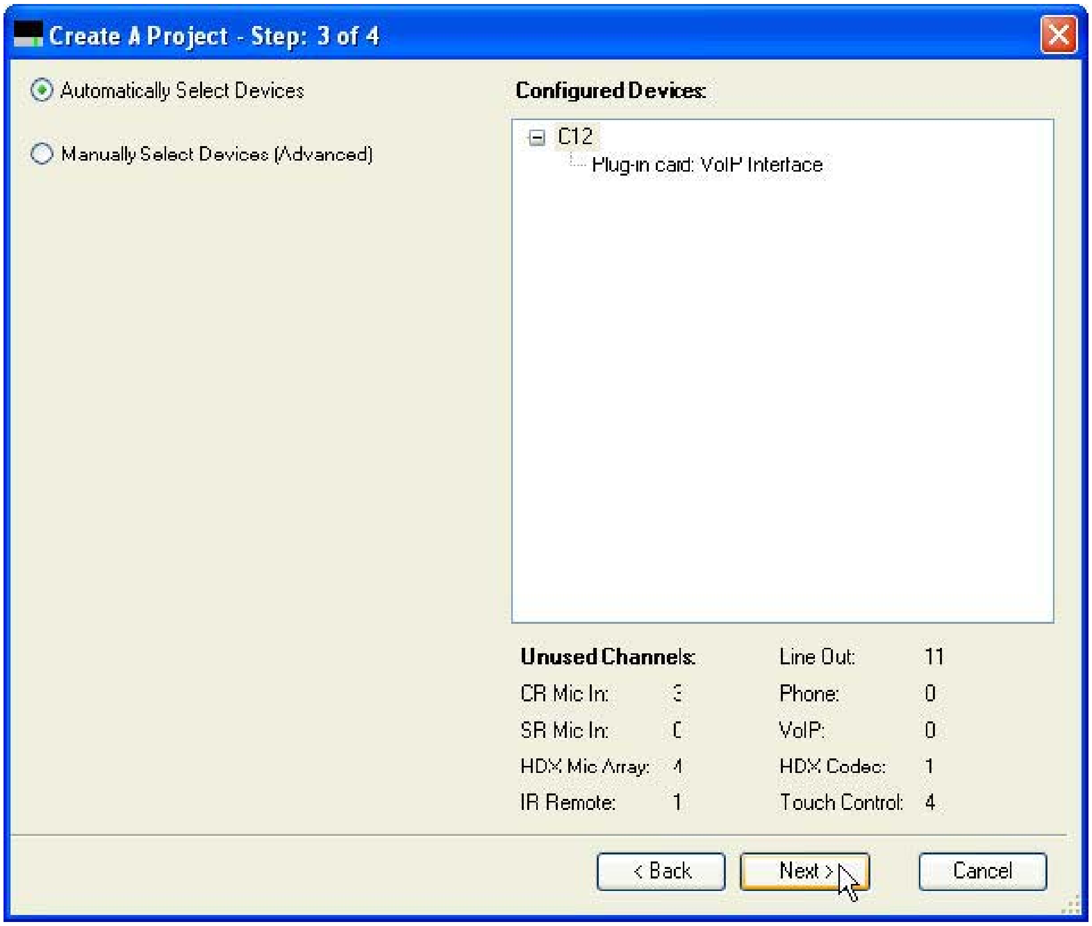
Quantity: 1

Label: Amplifier

A mono audio amplifier channel with consumer-grade (unbalanced) intputs driving one or more speakers.
Output Gain: -10 dB

Click **Next** to proceed to the next step.

5 In *Create a Project - Step 3,* select the SoundStructure equipment required to implement this project. By default SoundStructure Studio will select the lowest cost equipment that will meet the design requirements. In this example, the equipment required is a SoundStructure C12 and a SoundStructure VoIP Interface as shown in the following figure.



Create A Project - Step: 3 of 4

● Automatically Select Devices

○ Manually Select Devices (Advanced)

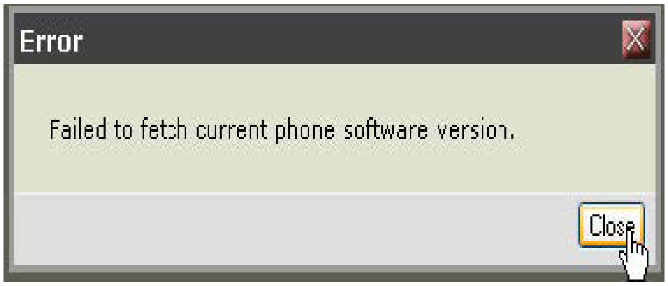
Configured Devices:

⊟ C12
  Plug-in card: VoIP Interface

Unused Channels:		Line Out:	11
CR Mic In:	3	Phone:	0
SR Mic In:	0	VoIP:	0
HDX Mic Array:	4	HDX Codec:	1
IR Remote:	1	Touch Control:	4

Click **Next** to continue to the next step.

6 In *Create A Project - Step 4,* you can either send the project directly to the SoundStructure system or to continue to work off line (default). Polycom recommends you continue to work offline and click **Finish** to complete the project and remain offline.

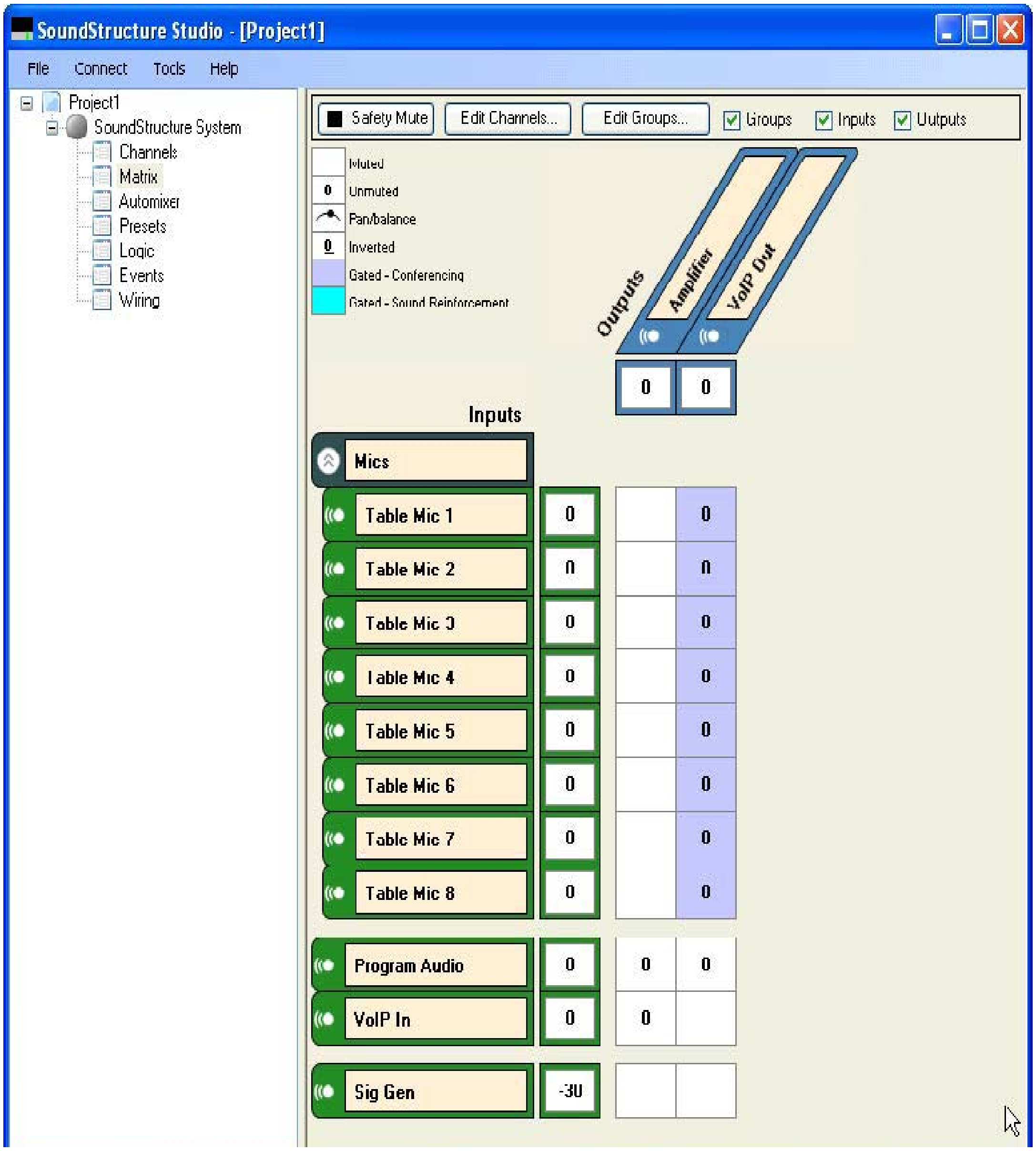


Error

Failed to fetch current phone software version.

The resulting project is shown in SoundStructure Studio. Select the **Matrix** tab to view the matrix as shown next.

Project Matrix

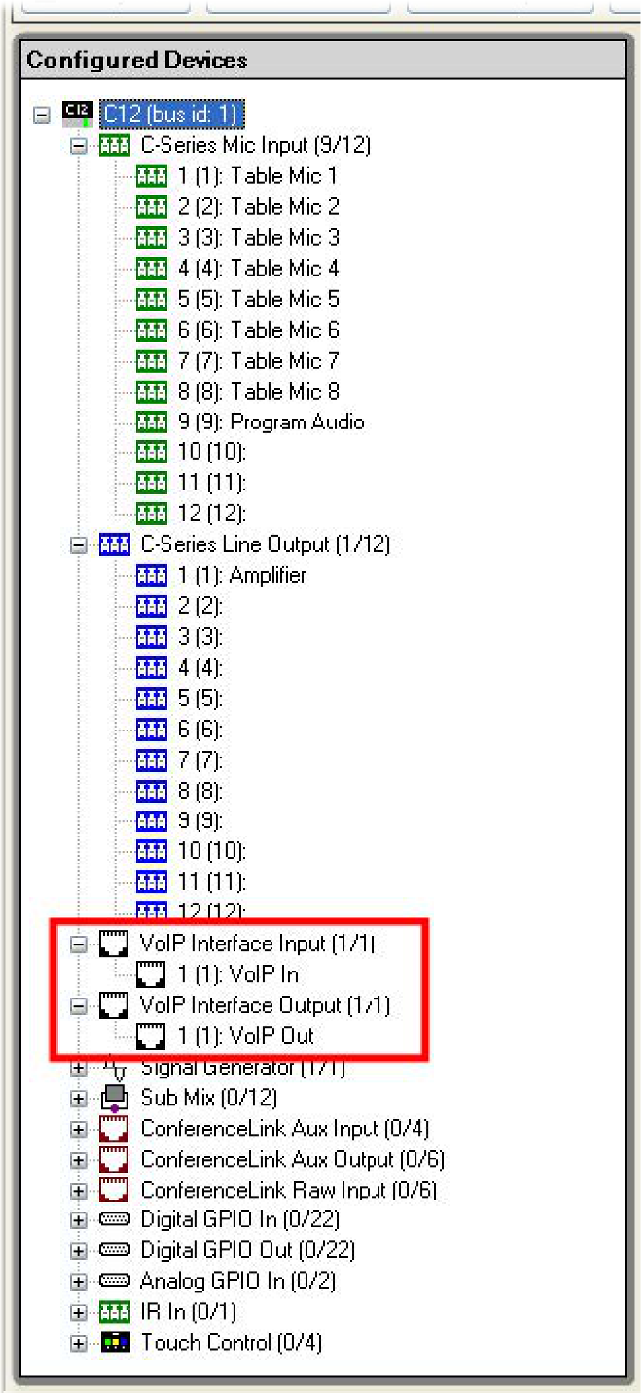

SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
Muted
Unmuted
Pan/balance
Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs

Inputs		Amplifier	VoIP Out
		0	0
Mics			
Table Mic 1	0		0
Table Mic 2	0		0
Table Mic 3	0		0
Table Mic 4	0		0
Table Mic 5	0		0
Table Mic 6	0		0
Table Mic 7	0		0
Table Mic 8	0		0
Program Audio	0	0	0
VoIP In	0	0	
Sig Gen	-30		

By default the audio from the VoIP In and Program Audio channels are routed to the Amplifier while the Program Audio and the echo canceled microphone audio are routed to the VoIP Out channel.

The wiring page of the new project will show the SoundStructure VoIP Interface installed as shown in the following figure.

Project Wiring Page



Configured Devices

- C12 (bus id: 1)
  - C-Series Mic Input (9/12)
    - 1 (1): Table Mic 1
    - 2 (2): Table Mic 2
    - 3 (3): Table Mic 3
    - 4 (4): Table Mic 4
    - 5 (5): Table Mic 5
    - 6 (6): Table Mic 6
    - 7 (7): Table Mic 7
    - 8 (8): Table Mic 8
    - 9 (9): Program Audio
    - 10 (10):
    - 11 (11):
    - 12 (12):
  - C-Series Line Output (1/12)
    - 1 (1): Amplifier
    - 2 (2):
    - 3 (3):
    - 4 (4):
    - 5 (5):
    - 6 (6):
    - 7 (7):
    - 8 (8):
    - 9 (9):
    - 10 (10):
    - 11 (11):
    - 12 (12):
  - VoIP Interface Input (1/1)
    - 1 (1): VoIP In
  - VoIP Interface Output (1/1)
    - 1 (1): VoIP Out
  - Signal Generator (1/1)
  - Sub Mix (0/12)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)
  - Touch Control (0/4)

At this point, you have created an offline project that includes the SoundStructure VoIP Interface. Configuration of the SoundStructure VoIP Interface settings will be described in Configuring the SoundStructure VoIP Interface.

If you have completed your customization, you can now upload the configuration file to a SoundStructure C12 system that has the SoundStructure VoIP Interface installed and continue with configuring the SoundStructure VoIP Interface settings once connected online.

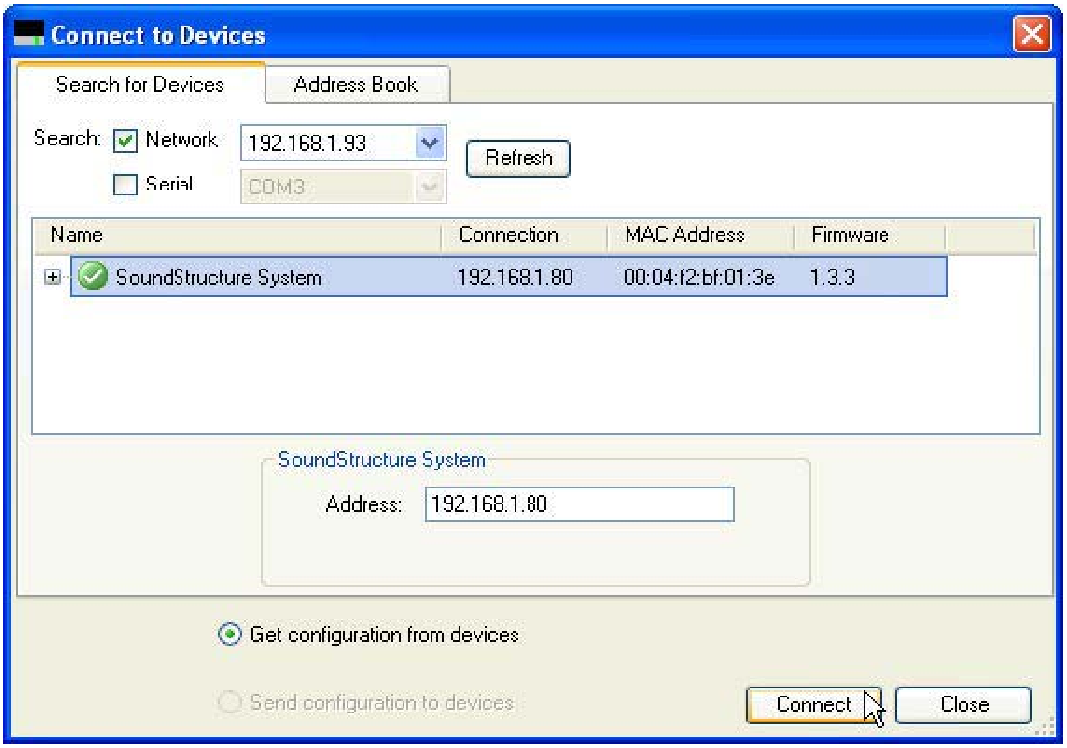
If you upload a configuration that requires a SoundStructure VoIP Interface to a SoundStructure system that does not have a SoundStructure VoIP Interface installed, the Upgrade Plug-in Card tool will be automatically run to allow you to modify the project to match the installed hardware.

Upgrading the Firmware in the SoundStructure System

Before loading a project with a SoundStructure VoIP Interface to a SoundStructure system, ensure that you are running SoundStructure device firmware 1.5.0 or later.

To upgrade the firmware in the SoundStructure system, use the following steps.

1. Select the **Connect** menu option in SoundStructure Studio and select the **Search for Devices** option and select the desired system from the list of discovered systems. Systems will be discovered if they are on the same subnet as your computer. If your system is not discovered, you can manually enter the IP address of the system or choose a system from the address book. You may also connect to your system over RS-232 if you don't have an IP connection.
2. Once the system is identified, select the **Get configuration from devices** option and select **Connect** as shown next.



Search: ☑ Network 192.168.1.93

☐ Serial COM3

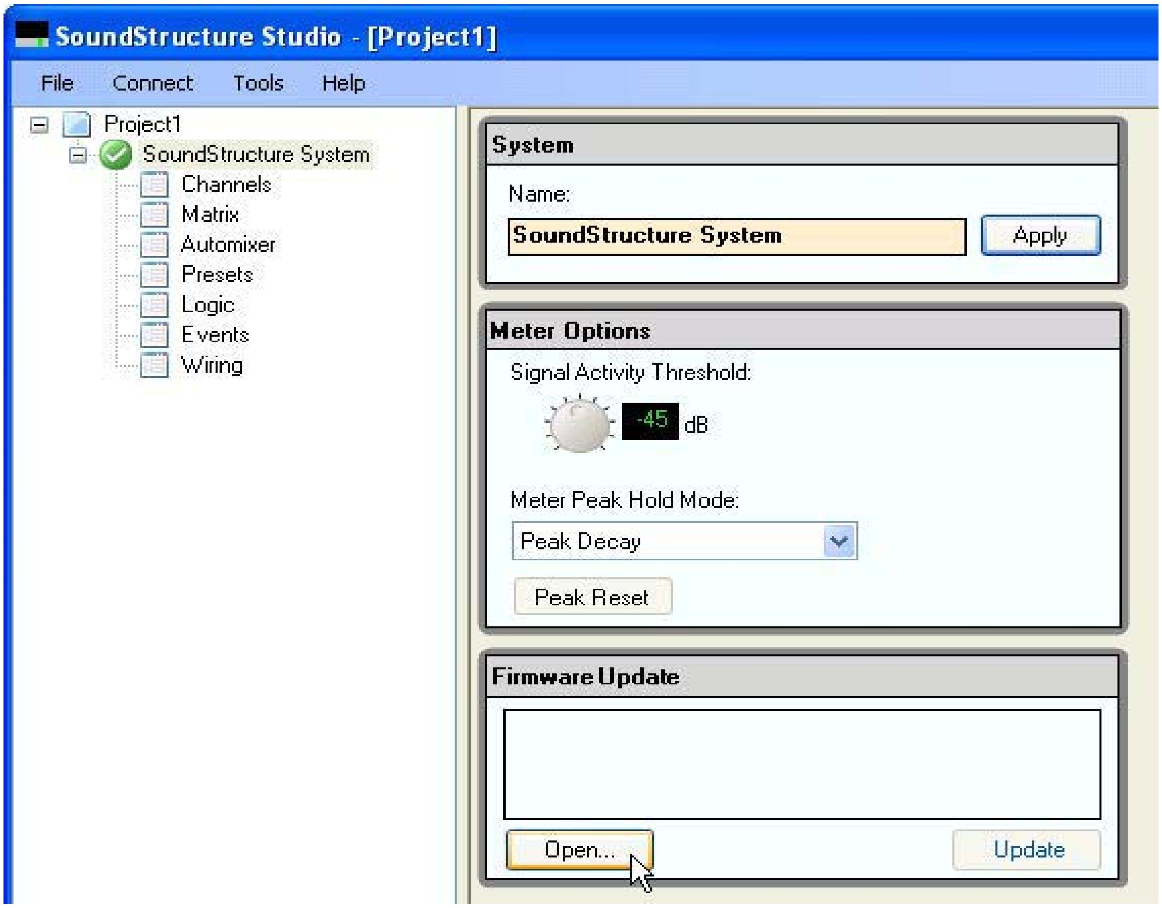
Name	Connection	MAC Address	Firmware
SoundStructure System	192.168.1.80	00:04:f2:bf:01:3e	1.3.3

Address: 192.168.1.80

● Get configuration from devices

○ Send configuration to devices

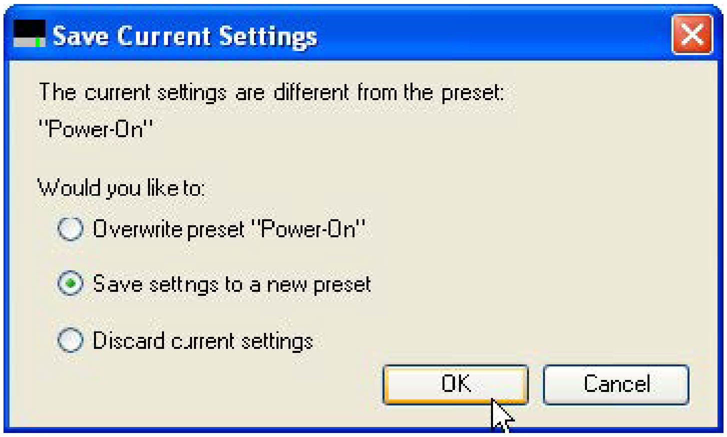
3 Once connected to the SoundStructure system, left click on the project name to get to the main system options page and select the **Open** button on the Firmware update control.


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring
System
Name:
SoundStructure System
Apply
Meter Options
Signal Activity Threshold:
-45
dB
Meter Peak Hold Mode:
Peak Decay
Peak Reset
Firmware Update
Open...
Update

4 Next, navigate to the desired SoundStructure device firmware file, and click **Update**. SoundStructure device firmware files have a bin file extension. Click **Yes** to confirm the firmware update when you are prompted.

5 If you are prompted to save your configuration file and you want to save the current settings, select **Yes**. If the settings have changed since the last preset was executed, you are prompted with the options of Overwrite preset Power-On, Save settings to a new preset, or Discard current settings. If you want to preserve any changes you've made to the preset that runs when the system is powered

on, select **Overwrite preset “Power-On”**. If you want to save your settings to a new preset, select **Save settings to a new preset.** If you don’t need to save your changes, select **Discard current settings**.


Save Current Settings

The current settings are different from the preset:
"Power-On"

Would you like to:
○ Overwrite preset "Power-On"
● Save settings to a new preset
○ Discard current settings

6 Once the firmware has been updated, the SoundStructure system will automatically reboot and SoundStructure Studio will display the Connect to Devices dialog. Close the Connect to Devices dialog and wait for the SoundStructure system to reboot. The SoundStructure front panel light will blink green while the system is booting and will turn solid green when the system has finished rebooting.

Installing the New Plugin Cards

Once the SoundStructure system has the appropriate firmware loaded, the next step is to power down the system and install (or replace) the plug-in cards with the newly selected cards.

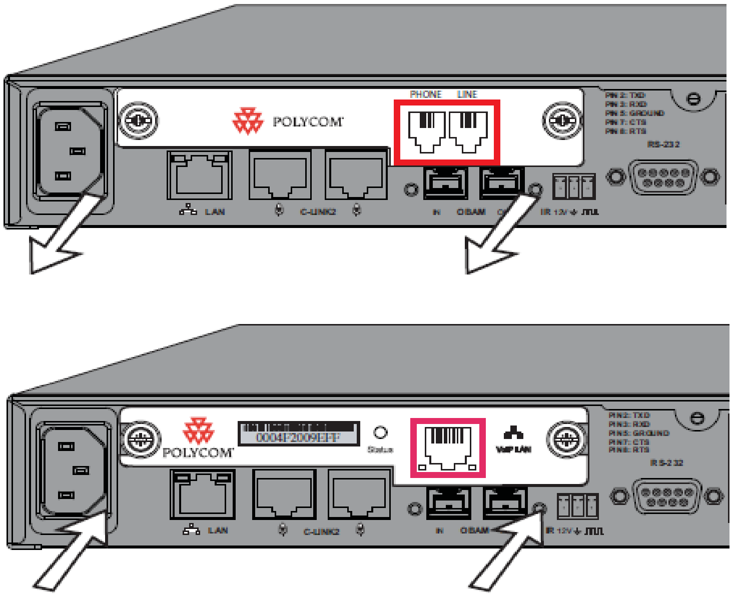
Power Down the System

To power down the system, remove the power cord from the rear of the SoundStructure system. If there are multiple devices in a system, remove power from all the devices.

Install Plug-in Cards

Remove any existing plug-in cards that were replaced with the Upgrade Plug-in Card Tool and replace with the new plug-in cards. The following figure shows a SoundStructure TEL1 being replaced with a SoundStructure VoIP Interface.

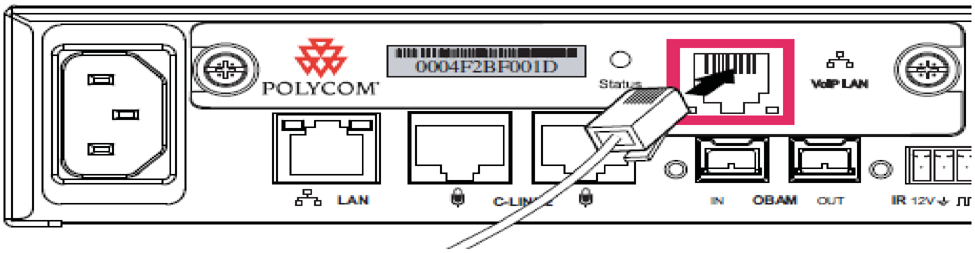
SoundStructure TEL1 Replaced with a SoundStructure VoIP Interface


PHONE
LINE
POLYCOM
LAN
C-LINK2
IN
OBAM
IR 12V
PIN 2: TXD
PIN 3: RXD
PIN 5: GROUND
PIN 7: CTS
PIN 8: RTS
RS-232
0004F2009EFF
Status
VoIP LAN

Connect Network to the SoundStructure VoIP Interface

Connect the network interface of the SoundStructure VoIP Interface to the appropriate VoIP network as shown in the following figure.

Connecting SoundStructure VoIP Interface to VoIP Network


POLYCOM
0004F2BF001D
Status
VoIP LAN
LAN
C-LINK2
IN
OBAM
OUT
IR 12V

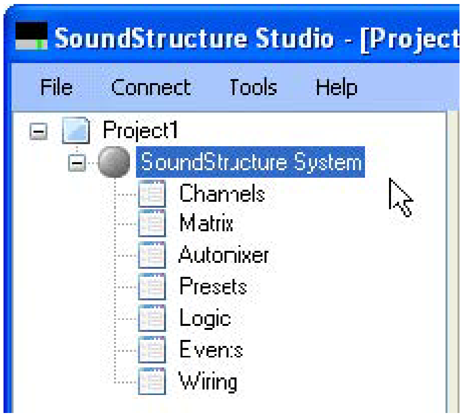
Power Up the System

Once the plug-in cards are properly installed, plug-in the power cord to the devices. Unless the system is in a factory-fresh state, after the system has finished booting, it will have a solid yellow front panel LED to indicate that the configuration file that is loaded does not match the hardware. To correct the yellow front panel LED status you will upload the proper configuration file in the next step.

Uploading the Configuration File

Now that the SoundStructure system has the appropriate plug-in cards installed, the upgraded project must be sent to the SoundStructure system with the following steps:

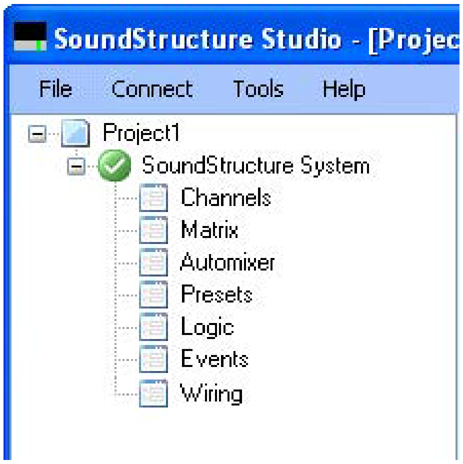
1. Left-click on the System Name to select the upgraded configuration file that was created previously.


SoundStructure Studio - [Project
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring

2. Connect to the SoundStructure system by selecting the **Connect** option and **Search for Devices**.
3. Select **Send configuration to devices** and then click **Connect.**
4. Confirm that you would like to send the configuration file to the system.

After sending the configuration file to the SoundStructure system, the resulting system will have a solid green front panel LED to indicate that the configuration file that is loaded and matches the hardware, and SoundStructure Studio project status will be solid green as shown next.

Green Status Indicator in SoundStructure Studio


SoundStructure Studio - [Projec
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring

The system is now ready for the final online setup required by the SoundStructure VoIP Interface to register with the desired call platform.

Configuring the SoundStructure VoIP Interface

Once the SoundStructure VoIP Interface is installed in the SoundStructure system, you need to configure the SoundStructure VoIP Interface for your call management environment with the following steps:

1. Set the IP address of the SoundStructure VoIP Interface. An IP address is required to access to the Web Configuration Utility.
2. Set the Provisioning Server settings. A provisioning server can be used to store the configuration parameters of the SoundStructure VoIP Interface and for upgrading the firmware on the SoundStructure VoIP Interface.

   If a provisioning server is not being used, the Web Configuration Tool must be used.
3. Configure the Call Server IP address so the SoundStructure VoIP Interface knows where to try to register its lines.
4. Register the line(s). The line registration information configures the extensions and authentication credentials required by the call server.

Details of these steps are in the following sections.

Working Online Only with SoundStructure VoIP Interface

The VoIP-specific settings described above can only be set while working online with the SoundStructure VoIP Interface.

Setting the IP address of the SoundStructure VoIP Interface

This section describes the default IP address settings of the SoundStructure VoIP Interface and how to set the IP address.

Understanding the Default Network Settings

The factory default values for the network settings of the SoundStructure VoIP Interface are:

- DHCP enabled which causes the interface to get an IP address from a DHCP server.
- Boot server configuration options are set to **Custom+Option 66**
- VLAN set to **dynamic**

If there is no DHCP server on the network, the SoundStructure VoIP Interface will not get an IP address. In this case, you will need to manually configure the network address by connecting to the SoundStructure system with SoundStructure Studio and navigating to the wiring page and selecting **Edit Network Settings**. See the section Setting an IP address with SoundStructure Studio for more information.

When the SoundStructure VoIP Interface is reset via a power cycle or reboot, the following events occur:

1. If a static IP address is not set, the SoundStructure VoIP Interface will request an IP address from the DHCP server.
2. Assuming the SoundStructure VoIP Interface gets an IP address, it will request provisioning server information if the DHCP options are set to the default value.
3. If a provisioning server is found, the interface will attempt to log into the provisioning server with the user-supplied credentials.

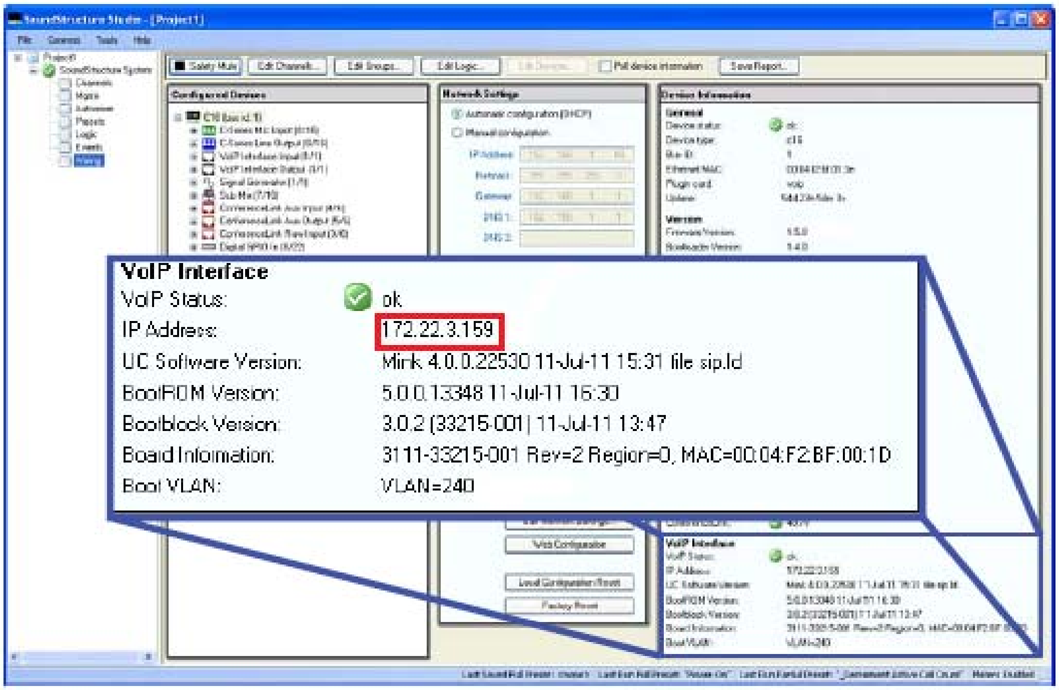
4. If the login to the provisioning server is successful, the appropriate bootROM and software files will be loaded if the files on the server are newer than the existing bootROM and software versions in the flash of the interface, and the appropriate configuration parameters will be loaded from the provisioning server.
5. If the interface is provisioned, it will use the SIP server registration information and appropriate credentials found from the device configuration parameters and register one or more lines with the call management server.
6. If the system SIP lines register properly, the interface will receive its line extension information and calls may be initiated.

Configuring the SoundStructure VoIP Interface Network Settings via the Web Configuration Utility

You can manually configure the SoundStructure VoIP Interface's IP address and provisioning settings with SoundStructure Studio or with the Web Configuration Utility.

To find the IP address of the SoundStructure VoIP Interface, connect to the online SoundStructure device with SoundStructure Studio and navigate to the **Wiring** page. The IP address of the SoundStructure VoIP Interface is found in the lower right-hand portion of the display as shown in the following figure.

SoundStructure VoIP Interface IP Address



VoIP Interface
VoIP Status: ✓ ok
IP Address: 172.22.3.159
UC Software Version: Mink 4.0.0.22530 11-Jul-11 15:31 file sip.ld
BootROM Version: 5.0.0.13348 11-Jul-11 16:30
Bootblock Version: 3.0.2 (33215-001 | 11-Jul-11 13:47
Board Information: 3111-33215-001 Rev=2 Region=0, MAC=00:04:F2:BF:00:1D
Boot VLAN: VLAN=240

In this example, the IP address of the SoundStructure VoIP Interface is 172.22.3.159.

If the network address is not listed, then the SoundStructure VoIP Interface did not receive an IP address from a DHCP server and either its network cable must be installed or its IP address must be configured manually through SoundStructure Studio.

Note: SoundStructure VoIP Interface Network Supporting RTP Packets

If you have a valid IP address for the SoundStructure VoIP Interface but can't browse into the interface with the Web Configuration Utility and can't network ping the interface, it is possible you have a network route to the interface that only supports RTP packets. Contact your network administrator to create a network route from your data network to your VoIP network that will allow you to configure the SoundStructure VoIP Interface.

Once you have a valid IP address for the SoundStructure VoIP Interface and have access to that particular network, you may customize the settings using the Web Configuration Utility.

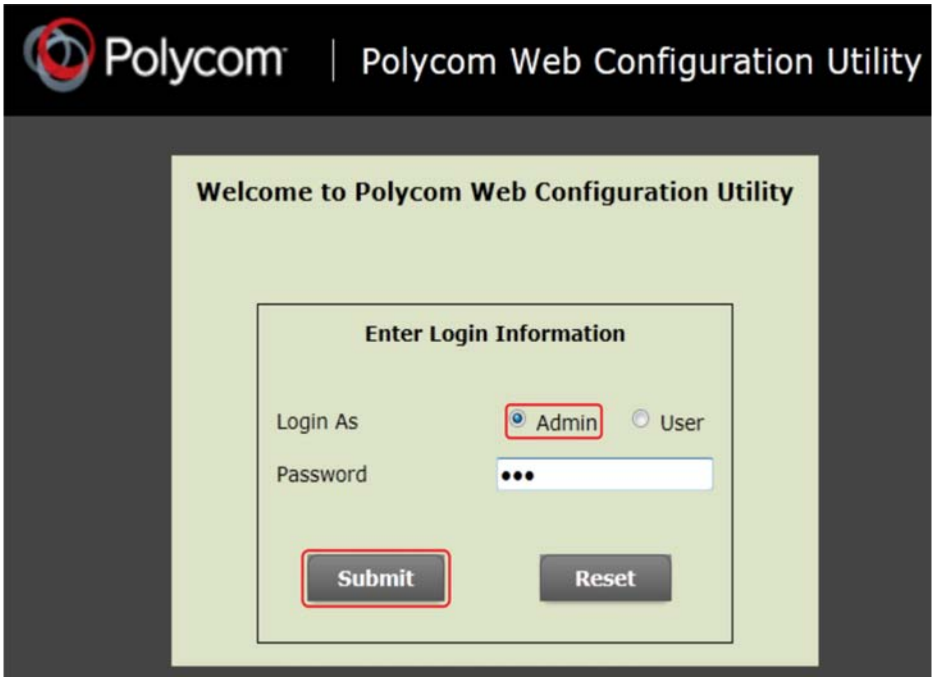
The Web Configuration Utility has context sensitive Field Help information that displays on the right side of the web page and provides detailed information on the parameter settings.

Web Configuration Utility login

Either click the *Web Configuration* button to open the default browser on the PC or enter the IP address of the SoundStructure VoIP Interface into your browser to start the *Web Configuration Utility*.

Once your browser window is opened, you will be presented with a login prompt as shown in the following figure. Select **Admin** and use the default password of **456**.

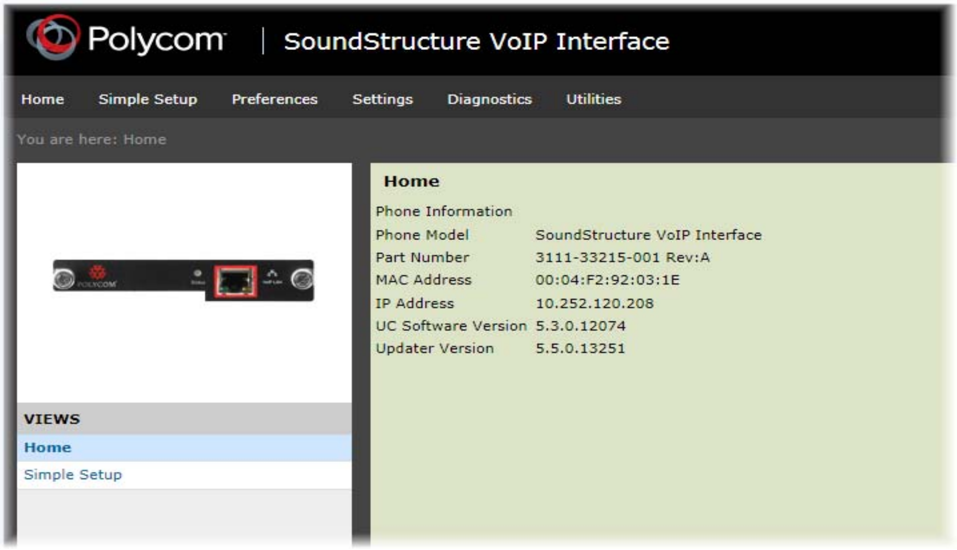
Web Configuration Utility


Polycom | Polycom Web Configuration Utility
Welcome to Polycom Web Configuration Utility
Enter Login Information
Submit
Reset

Login As ● Admin ○ User

Password •••

Once logged in, you are presented with a series of settings including the model number, part number, MAC address, IP address, UC Software version, and Updater software version as shown next.


Polycom
SoundStructure VoIP Interface
Home
Simple Setup
Preferences
Settings
Diagnostics
Utilities
You are here: Home
VIEWS
Home
Simple Setup

Home

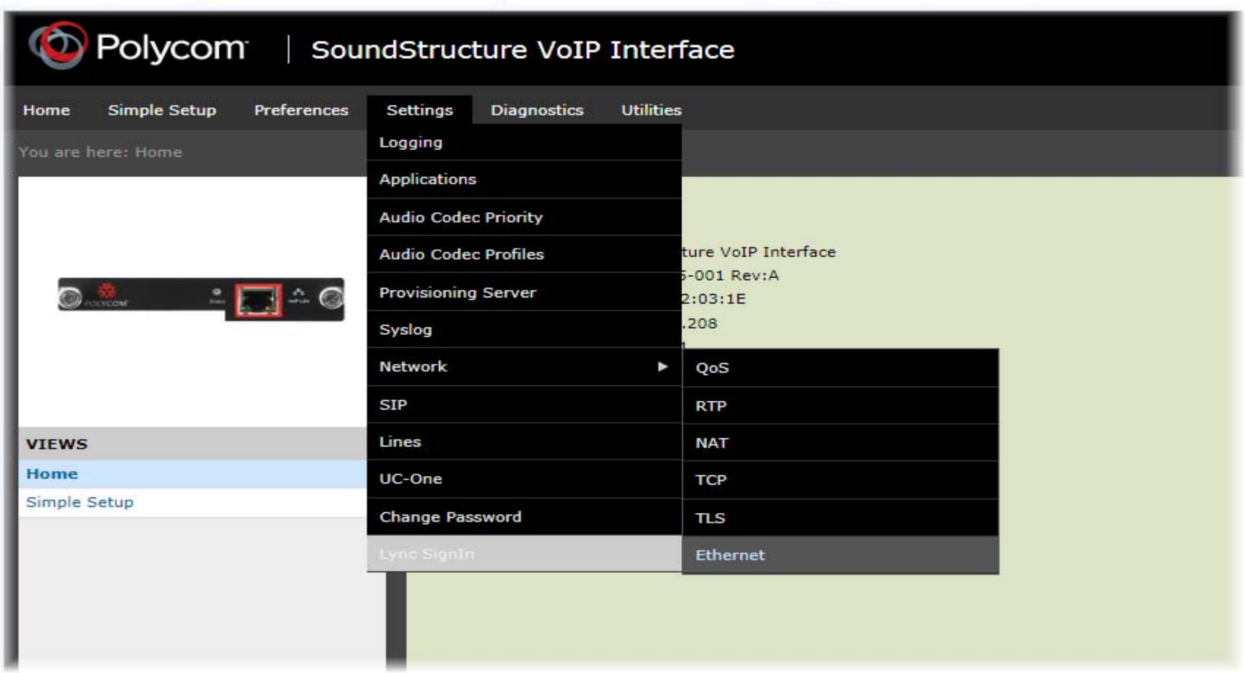
Phone Information

Phone Model	SoundStructure VoIP Interface
Part Number	3111-33215-001 Rev:A
MAC Address	00:04:F2:92:03:1E
IP Address	10.252.120.208
UC Software Version	5.3.0.12074
Updater Version	5.5.0.13251

Setting the Ethernet Address

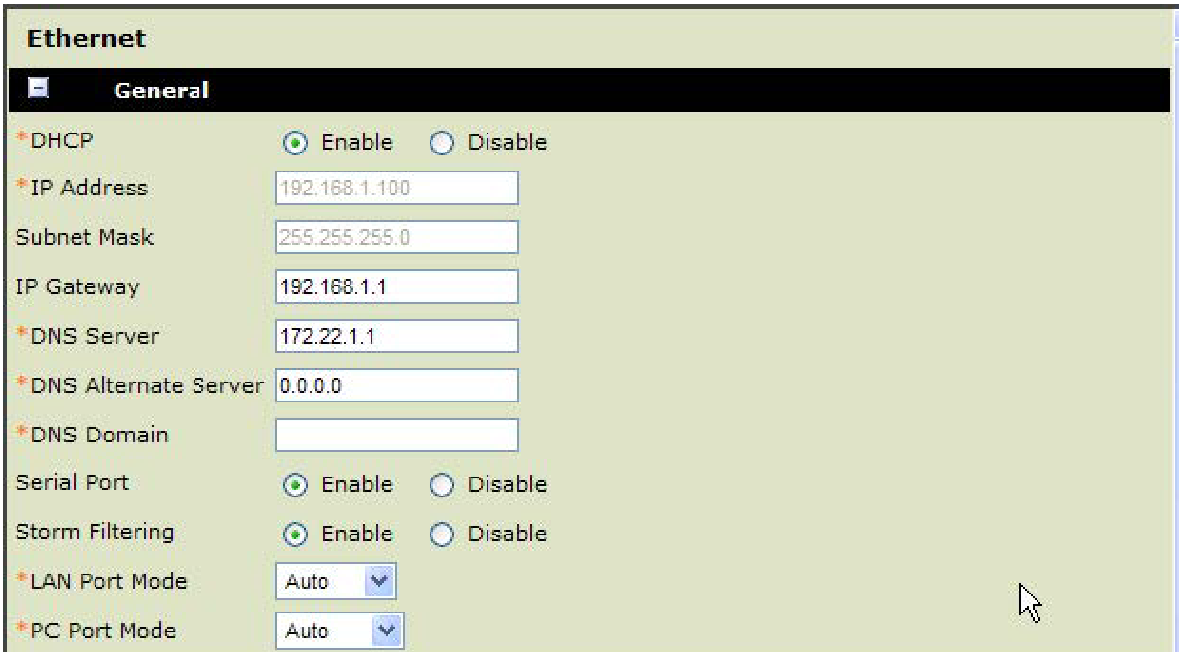
To customize the Ethernet settings of SoundStructure VoIP Interface, navigate to the Ethernet selection from the **Settings** > **Network** menu as shown next.

Setting Ethernet Address


Polycom
SoundStructure VoIP Interface
Home
Simple Setup
Preferences
Settings
Diagnostics
Utilities
You are here: Home
Logging
Applications
Audio Codec Priority
Audio Codec Profiles
Provisioning Server
Syslog
Network
SIP
Lines
UC-One
Change Password
Lync SignIn
QoS
RTP
NAT
TCP
TLS
Ethernet
ture VoIP Interface
5-001 Rev:A
2:03:1E
.208
VIEWS
Home
Simple Setup

The Ethernet settings page displays as shown next. The fields shown are a superset of the fields available via SoundStructure Studio described earlier in Setting an IP address with SoundStructure Studio.

Ethernet Settings Page



Ethernet

General

*DHCP ● Enable ○ Disable

*IP Address 192.168.1.100

Subnet Mask 255.255.255.0

IP Gateway 192.168.1.1

*DNS Server 172.22.1.1

*DNS Alternate Server 0.0.0.0

*DNS Domain

Serial Port ● Enable ○ Disable

Storm Filtering ● Enable ○ Disable

*LAN Port Mode Auto

*PC Port Mode Auto

The General fields are described in the following table.

General Settings

Name	Possible Values	Description
DHCP	Enable (default) or Disable	If disabled, you must enter the rest of the IP parameters in order to have a valid IP address.
IP Address	Dotted-decimal IP address	Enter the IP address to be used for the SoundStructure VoIP Interface. This IP address should be unique to the SoundStructure VoIP Interface. If DHCP is enabled, this setting will be grayed out.
Subnet Mask	Dotted-decimal subnet mask	Enter the subnet netmask to be used with the IP address. Typically this is equal to 255.255.255.0. If DHCP is enabled, this setting will be grayed out.
IP Gateway	Dotted-decimal IP address	Enter the IP address of the router that is the address the SoundStructure VoIP Interface will go to when seeking IP addresses outside of the local subnet.
DNS Server	Dotted-decimal IP address	Enter the IP address of the primary domain name server.
DNS Alternate Server	Dotted-decimal IP address	Enter the IP address of the secondary domain name server.
DNS Domain	Domain name string	The phone's Domain Name System

General Settings

Name	Possible Values	Description
Serial Port	Enables (default) or disables	If enabled, a debug serial port is active.
Storm Filtering	Enable (default) or Disable	If enabled, the DoS storm prevention state Ethernet packet filtering is used to prevent TCP/IP stack overflows caused by invalid or excessive data.
LAN Port Mode	Auto (Default), 10HD, 10FD, 100HD, 100FD, 1000FD	Choose the speed of the network on the Ethernet interface.

See the Polycom UC Software Administrators Guide 4.0.1 for additional information about the advanced settings of Ethernet 802.1x and PAC File Info.

The VLAN settings are described in the following table.

VLAN Settings

Name	Possible Values	Description
VLAN	Null, 0 through 4095	Enter the DHCP private option value to be used when VLAN discovery is set to custom
VLAN Filtering	Enable or Disable (default).	Enables or disables the VLAN Ethernet packet filtering on the phone to prevent TCP/IP stack overflows caused by invalid or excessive data.
LLDP	Enable (Default) or Disable.	If enabled, the phone will use the LLDP protocol to communicate with the network switch for certain network parameters. Most often this will be used to set the VLAN that the phone should use for voice traffic. It also reports the power management requirements to the switch.
CDP Compatibility	Enable (Default) or Disable.	If enabled, the phone will use CDP-compatible signaling to communicate with the network switch for certain network parameters. Most often this will be used to set the VLAN that the phone should use for voice traffic, and for the phone to communicate its PoE power requirements to the switch.
DHCP VLAN Discovery	Disabled, Fixed (default), or Custom.	If set to disabled, no VLAN discovery through DHCP. If set to Fixed, use predefined DHCP vendor-specific option values of 128, 144, 157, and 191. If one of these is used, VLAN Option is ignored. If set to Custom, use the number specified for the VLAN Option as the DHCP private option value.
DHCP VLAN Option	128 to 254.	The DHCP private option (when VLAN Discovery is set to Custom).

Note: Disabling CDP in Cisco Call Environments

In Cisco call management environments, you may need to disable the CDP Compatibility option found under **Settings** > **Network** > **Ethernet** to prevent the SoundStructure VoIP Interface from joining a Cisco VoIP-specific subnet.

Setting the Provisioning Server settings

Once the IP address of the SoundStructure VoIP Interface has been set, the next step is to configure the Provisioning Server settings if a provisioning server is to be used for accessing the VoIP configuration settings. If a provisioning server is not to be used, the VoIP configuration settings may be set manually using the Web Configuration Utility.

Full technical details concerning the SIP setting for the SoundStructure VoIP Interface may be found in the Polycom UC Software Administrators Guide 4.0.1. Additional detail for using the Web Configuration Utility can be found in the Polycom Web Configuration Utility User Guide.

Note: Central Provisioning the SoundStructure VoIP Interface Environment

Polycom recommends using a central provisioning server when setting up your VoIP environment with many phones. This allows for flexibility in installing, upgrading, maintaining, and configuring the SoundStructure VoIP Interface. Configuration, log, and directory files are normally located on this server. Polycom recommends giving the phone write-access to the server to support uploading logs from the phones.

If the SoundStructure VoIP Interface cannot locate a provisioning server when it boots up, it will operate with internally saved parameters. This is useful when the provisioning server is not available.

Using a Central Provisioning Server

You can centrally provision SoundStructure VoIP Interfaces from a provisioning server through a system of global configuration files and SoundStructure VoIP Interface-specific configuration files system. The central provisioning method uses the MAC address of the SoundStructure VoIP Interface to specify the set of configuration files to use from the provisioning server.

The provisioning server facilitates automated application upgrades, logging, and fault tolerance. To improve reliability, you can configure multiple redundant provisioning servers. Parameters can be stored in the files in any order and can be placed in any number of files. For example, it might be desirable to set the default audio codec for a remote user differently than for office users. By adding the audio codec settings to a particular user’s per-phone file, the values in the broader system file are ignored.

Using Manual Provisioning

You can manually configure the SoundStructure VoIP Interface by using the Web Configuration Utility. Any changes you make are stored in a configuration override file that will override any configuration settings that may been performed with a provisioning server. The override file is stored on the SoundStructure VoIP Interface, but a copy is also uploaded to the central provisioning server (if one is being used). When the SoundStructure VoIP Interface boots, the SoundStructure VoIP Interface software loads the override file. The settings in this file override the settings in the centrally provisioned files.

Understanding the Hierarchy of SoundStructure VoIP Interface Settings

The VoIP parameters can be configured by files on a provisioning server, from the Web Configuration Utility, or from SoundStructure Studio. Because there are multiple ways to set these parameters there is a hierarchy to which settings are used: the SIP settings from the central provisioning server are used unless overridden by a higher priority setting from the Web Configuration Utility settings.

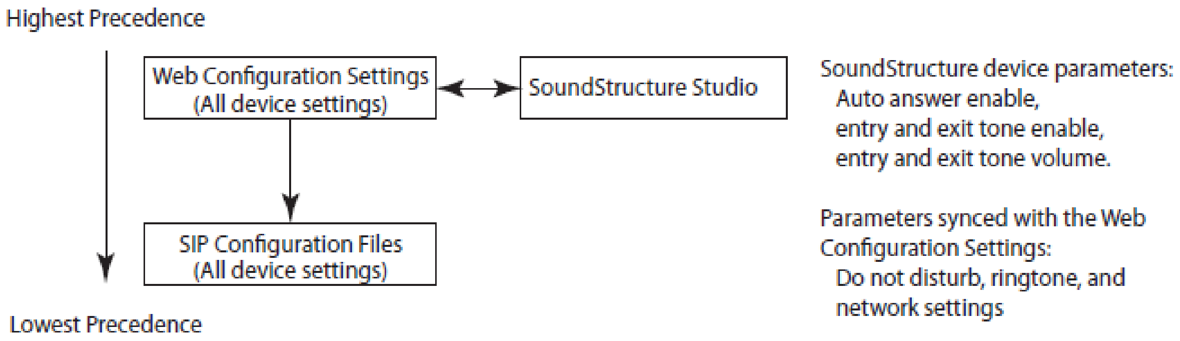
SoundStructure Studio provides an interface for a small subset of the overall SoundStructure VoIP Interface settings including network settings and room-based settings such as auto answer enable, ringtone selection, entry and exit tones, and the tone volume. All other settings are configured via the Web Configuration Utility or via configuration files on a central provisioning server. When SoundStructure VoIP Interface parameters are set within SoundStructure Studio those parameters are synchronized with the web interface settings and stored in the SoundStructure VoIP Interface. SoundStructure-specific telephony parameters such as auto answer, entry and exit tones, and tone volumes are stored within the SoundStructure device configuration file.

If you are not using a provisioning server, Polycom recommends using the Web Configuration Utility for setting the VoIP parameters.

If you don’t have a valid network route to the Web Configuration Utility, use SoundStructure Studio to set the IP address of the SoundStructure VoIP Interface as described in Setting an IP address with SoundStructure Studio.

The following figure shows that the SoundStructure Studio settings are synced with the settings stored in the SoundStructure VoIP Interface and that any settings made via the Web Configuration Utility override any corresponding settings made from configuration files on a provisioning server.

SoundStructure Studio Settings Synced with SoundStructure VoIP Interface Settings


Highest Precedence
Web Configuration Settings
(All device settings)
SoundStructure Studio
SoundStructure device parameters:
Auto answer enable,
entry and exit tone enable,
entry and exit tone volume.
SIP Configuration Files
(All device settings)
Parameters synced with the Web
Configuration Settings:
Do not disturb, ringtone, and
network settings
Lowest Precedence

Configuring Provisioning Server Settings via the Web Configuration Utility

A provisioning server can be used to store the configuration files for the SoundStructure VoIP Interface, store the log files from the interfaces, and store firmware files for upgrading the SoundStructure VoIP Interface.

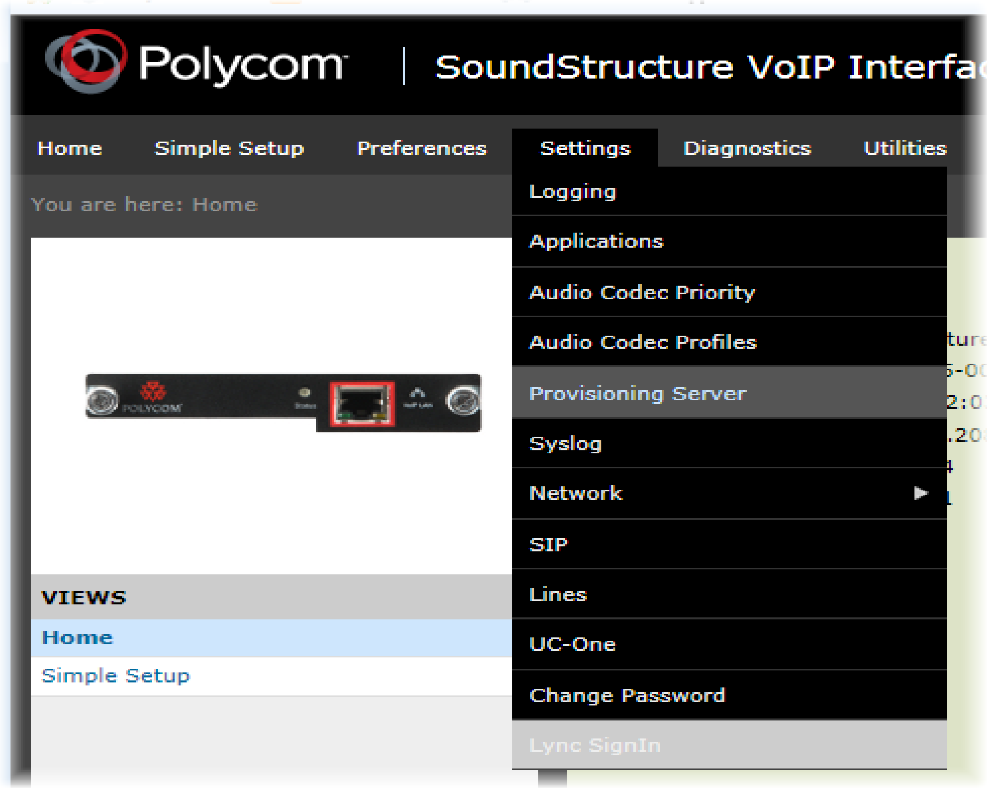
The configuration files provide the information required for the SoundStructure VoIP Interface to successfully access the call management platform and to customize the behavior of the interface.

You can use the Web Configuration Utility to configure how the SoundStructure VoIP Interface accesses the provisioning server.

Setting the Provisioning Server Information

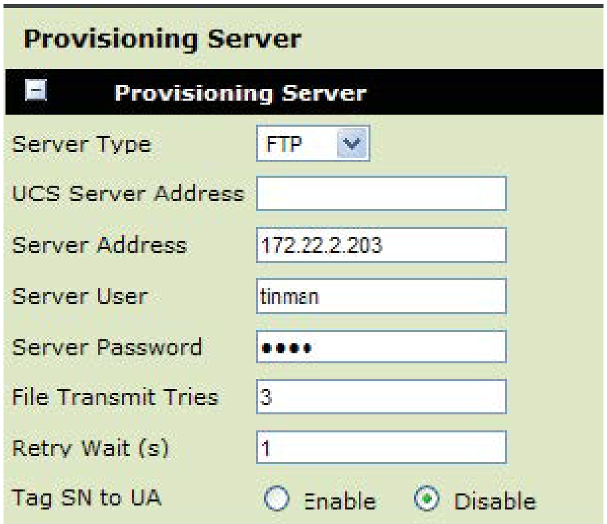
To configure the provisioning server information, navigate to the Provisioning selection from the Settings menu as shown next.

Setting Up the Provisioning Server


Polycom
SoundStructure VoIP Interfa
Home
Simple Setup
Preferences
Settings
Diagnostics
Utilities
You are here: Home
Logging
Applications
Audio Codec Priority
Audio Codec Profiles
Provisioning Server
Syslog
Network
SIP
Lines
UC-One
Change Password
Lync SignIn
VIEWS
Home
Simple Setup

Enter the provisioning server information settings as shown next. These settings are a superset of the settings that can be configured directly with SoundStructure Studio on the Wiring page.

Provisioning Server Page



Provisioning Server

Provisioning Server

Server Type	FTP
UCS Server Address	
Server Address	172.22.2.203
Server User	tinman
Server Password	••••
File Transmit Tries	3
Retry Wait (s)	1
Tag SN to UA	○ Enable ● Disable

Setting the provisioning server settings manually requires setting the *DHCP server type* to Static, otherwise the DHCP server provided provisioning server settings will be used. The fields are described in the following table.

Provisioning Server Settings

Name	Possible Values	Description
Server Type	FTP, TFTP, FTPS, HTTP, HTTPS	This configuration identifies the provisioning server the SoundStructure VoIP Interface downloads software and configurations from as well as to where the SoundStructure VoIP Interface uploads logs and configuration files.
UCS Server Address	Maximum of 256 characters	Enter the URL of the download server that the SoundStructure VoIP Interface uses to obtain software and upgrades.
Server Address	Maximum of 256 characters	Use this provisioning server if the DHCP client is disabled, if the DHCP server does not send a boot server option, or if the boot server parameter is set to Static. If using a URL, you can supply a user name and password.
Server User	Maximum of 256 characters	The user name required for the SoundStructure VoIP Interface to log in to the provisioning server (if required).
Server Password	Maximum of 256 characters	The password required for the SoundStructure VoIP Interface to log in to the provisioning server (if required).
File Transmit Tries	1 to 10	This setting specifies the number of times to attempt a file transfer. Choose a value from 1 to 10. The default is 3.

Provisioning Server Settings

Name	Possible Values	Description
Retry Wait (s)	0 to 300 seconds	This setting specifies the minimum amount of time that must elapse between starting a new file transfer and retrying a file transfer. You can specify a value from 0 to 300 seconds. The default is 1 second.
Tag SN to UA	Enable or Disable	This setting specifies whether the SoundStructure VoIP Interface's serial number (MAC address) is included in the User-Agent header of any HTTP or HTTPS provisioning request. When enabled, the MAC address is present, when disabled, it is not. The default is disabled.

The DHCP boot server settings are only accessible when the DHCP client is enabled. The Boot server parameters are described in the following table.

DHCP Boot Server Settings

Name	Possible Values	Description
Boot server	Static (default)	The SoundStructure VoIP Interface will use the boot server/provisioning server configured manually through the Server options.
	Custom	The SoundStructure VoIP Interface will look for the option number specified by the boot server option parameter and the type specified by the boot server option type in the response received from the DHCP server. If the DHCP server sends nothing, the following scenarios are possible: • If a boot server value is stored in flash memory and the value is not 0.0.0.0, then the value stored in flash is used. • Otherwise the SoundStructure VoIP Interface sends out a DHCP INFORM query. If a single alternate DHCP server responds, this is functionally equivalent to the scenario where the primary DHCP server responds with a valid boot server value. If no alternate DHCP server responds, the INFORM query process will retry and eventually time out.

DHCP Boot Server Settings

Name	Possible Values	Description
Boot server	Option 66	The SoundStructure VoIP Interface will look for Option number 66 (string type) in the response received from the DHCP server. The DHCP server should send address information in Option 66 that matches one of the formats described for Server Address. If the DHCP server sends nothing, the following scenarios are possible: • If a boot server value is stored in flash memory and the value is not 0.0.0.0, then the value stored in flash is used. • Otherwise the SoundStructure VoIP Interface sends out a DHCP INFORM query. If a single alternate DHCP server responds, this is functionally equivalent to the scenario where the primary DHCP server responds with a valid boot server value. If no alternate DHCP server responds, the INFORM query process will retry and eventually time out.
	Custom + Option 66	The SoundStructure VoIP Interface will first use the custom option if present or use Option 66 if the custom option is not present. If the DHCP server sends nothing, the following scenarios are possible: • If a boot server value is stored in flash memory and the value is not 0.0.0.0, then the value stored in flash is used. • Otherwise the SoundStructure VoIP Interface sends out a DHCP INFORM query. If a single alternate DHCP server responds, this is functionally equivalent to the scenario where the primary DHCP server responds with a valid boot server value. If no alternate DHCP server responds, the INFORM query process will retry and eventually time out.
Boot Server Option	128 through 254	When the boot server parameter option is set to Custom, this parameter specifies the DHCP option number in which the SoundStructure VoIP Interface will look for its boot server.
Boot Server Type	IP Address	When the Boot Server parameter is set to Custom, this parameter specifies the type of DHCP option in which the SoundStructure VoIP Interface will look for its boot server. The IP Address must specify the boot server.
	String	The string must match one of the formats described for server address.

DHCP Boot Server Settings

Name	Possible Values	Description
Option 60 Format	RFC3925 binary	Vendor identifying information in the format defined in RFC 3925 which can be found at: Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4. For more information, refer to Technical Bulletin 54041: Using DHCP Vendor Identifying Options With Polycom Phones. **Note**: DHCP option 125 containing the RFC 3295 formatted data will be sent whenever option 60 is sent. **Note**: DHCP option 43 data is ignored.
	ASCII string	Vendor identifying information in ASCII. **Note**: DHCP option 125 containing the RFC 3295 formatted data will be sent whenever option 60 is sent. **Note**: DHCP option 43 data is interpreted as encapsulated DHCP options and these will take precedence over options received outside of option 43.

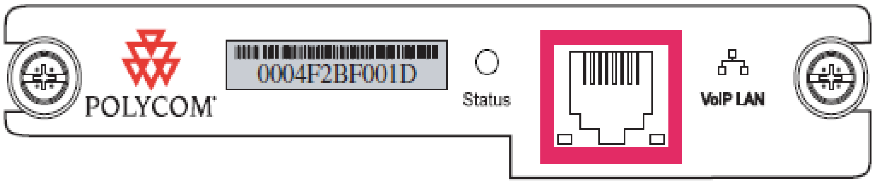
To save the settings select the save button on the bottom of the screen. Your changes may be viewed by selecting **View Modifications** prior to selecting **Save**.

Registering Lines with the SoundStructure VoIP Interface

To register one or more lines of the SoundStructure VoIP Interface with the call management platform, you need to supply the IT/Phone system administrator with the MAC address of the SoundStructure VoIP Interface and in return the IT/Phone system administrator will provide the SIP server IP address and line registration information including login credentials required to register with to the SIP server.

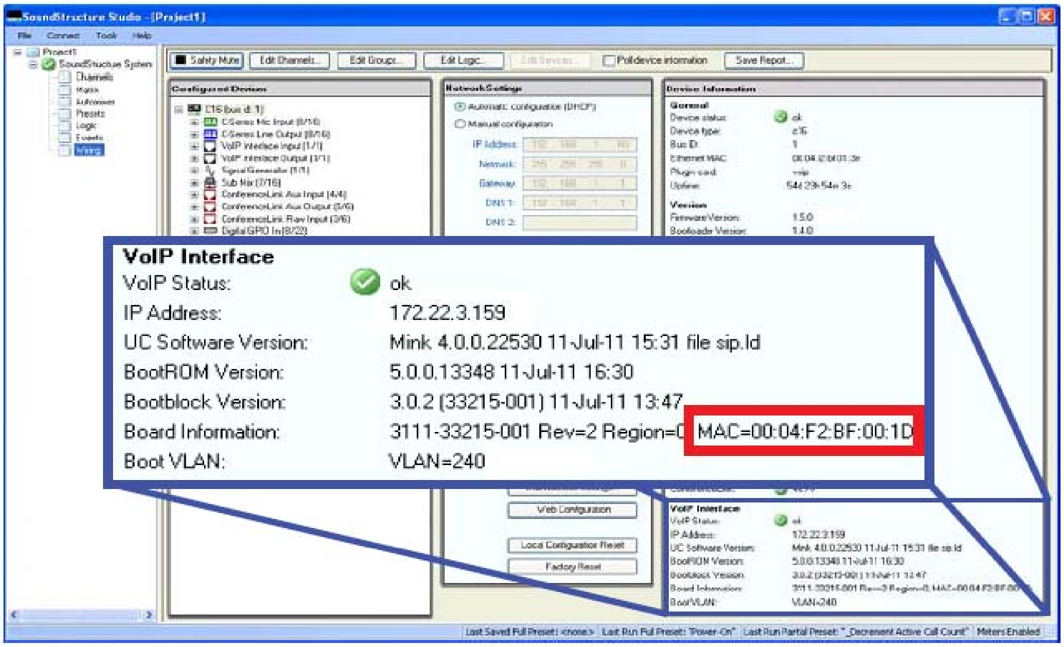
You can locate the MAC address on the rear of the SoundStructure VoIP Interface as shown next. In this example, the MAC address is `0004F2BF001D`.

SoundStructure VoIP Interface MAC Address



If the SoundStructure VoIP Interface is already installed in an equipment rack and not easily accessible, you may find the MAC address of the SoundStructure VoIP Interface on the Wiring page within SoundStructure Studio when connected to a SoundStructure system as shown next.

SoundStructure VoIP Interface MAC Address in SoundStructure Studio


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Autimixer
Presets
Logic
Events
Wiring
Safety Mute
Edit Channels...
Edit Groups...
Edit Logic...
Edit Devices...
Poll device information
Save Report...
Configured Devices
C16 (bus id: 1)
C-Series Mic Input (8/16)
C-Series Line Output (8/16)
VoIP Interface Input (1/1)
VoIP Interface Output (1/1)
Signal Generator (1/1)
Sub Mix (7/16)
ConferenceLink Aux Input (4/4)
ConferenceLink Aux Output (5/6)
ConferenceLink Raw Input (3/6)
Digital GPIO In (8/22)
Network Settings
Automatic configuration (DHCP)
Manual configuration
IP Address:
Netmask:
Gateway:
DNS 1:
DNS 2:
Web Configuration
Local Configuration Reset
Factory Reset
Device Information
General
Device status:
Device type:
Bus ID:
Ethernet MAC:
Plug-in card:
Uptime:
Version
Firmware Version:
Bootloader Version:
Last Saved Full Preset: <none>
Last Run Full Preset: "Power-On"
Last Run Partial Preset: "_Decrement Active Call Count"
Meters Enabled

VoIP Interface

VoIP Status: ✓ ok

IP Address: 172.22.3.159

UC Software Version: Mink 4.0.0.22530 11-Jul-11 15:31 file sip.ld

BootROM Version: 5.0.0.13348 11-Jul-11 16:30

Bootblock Version: 3.0.2 (33215-001) 11-Jul-11 13:47

Board Information: 3111-33215-001 Rev=2 Region=0 MAC=00:04:F2:BF:00:1D

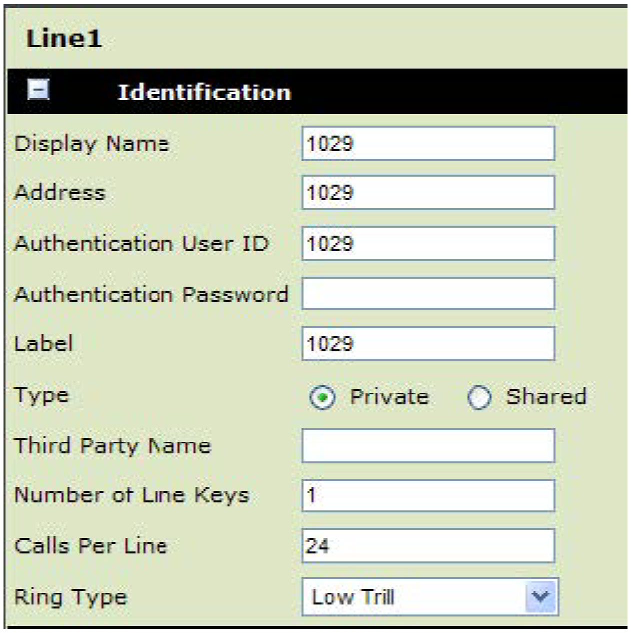
Boot VLAN: VLAN=240

If you are not using a provisioning server, you can manually setup one or more line registrations for the SoundStructure VoIP Interface with the Web Configuration Utility.

Configuring a Line Registration

To configure a line registration, navigate to the **Settings** > **Lines** option and select a line. The line settings appear as shown next.

Line Registration



Line1

Identification

Display Name	1029
Address	1029
Authentication User ID	1029
Authentication Password	
Label	1029
Type	● Private ○ Shared
Third Party Name	
Number of Line Keys	1
Calls Per Line	24
Ring Type	Low Trill

Enter the information as described in the table below.

Identification Settings

Name	Possible Values	Description
Display Name	Any string up to 256 characters	The display name used in SIP signaling that your phone uses as the default caller ID. This name is displayed on the call recipient's phone.
Address	Any string up to 256 characters	Enter a line identification address that the phone uses to register with the server. The address may include a user name, or the host of the phone's SIP URI. For example, if the phone's line is 1002@polycom.com, enter 1002 as the SIP where polycom.com is the server. Or, you can enter 1002@polycom.com. Any address entered will be displayed as the phone's line if the display name and label are not specified.
Authentication User ID	Any string up to 256 characters	Enter the user name used to authenticate this line registration (if applicable).
Authentication Password	Any string up to 256 characters	Enter the password used to authenticate this line registration (if applicable).

Identification Settings

Name	Possible Values	Description
Label	Any string up to 256 characters	Enter the text that will display next to the associated line key. If a label isn't defined, the label will be derived from the user part of the address.
Type	Private or Shared	Choose Private or Shared line identification. If set to Private, standard call signaling is used. If set to Shared, call state subscriptions and notifications are shared with multiple phones. The default is Private.
Third Party Name	Any string up to 256 characters	Enter the line identification number you want to use for this Bridged Line Appearance (BLA). This field is available only for BLA registration. You must set Type to Shared to register a BLA line.
Number of Line Keys	From 1 to 65536	The number of line keys that will be associated with this line registration. The default is 1.
Calls Per Line	From 0 to 24	The number of calls which may be active or on hold for each line key associated with this line registration.
Ring Type	**1** Default **2** Silent Ring **3** Low Trill **4** Low Double Trill **5** Medium Trill **6** High Trill **7** High Double Trill **8** Highest Trill **9** Highest Double Trill **10** Beeble **11** Triplet **12** Ringback-style **13** Low Trill Precedence **14** Ring Splash	Choose a specific ringtone to identify calls to this line.

Configuring a Call Server

To configure the primary call server, continue to the Server 1 settings and enter the appropriate parameters as described in the following table.

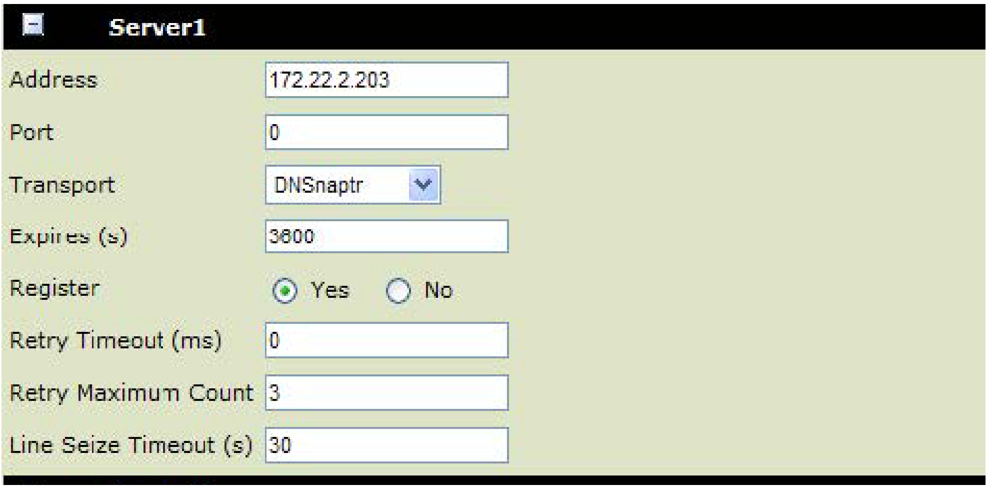
Call Server Setting

Name	Possible Values	Description
Address	Any string up to 256 characters	Enter an IP address or the host name of an SIP server that accepts registrations. This address can be set at Simple Setup > SIP Server, or at Settings > SIP > Server 1. Changes applied to settings in one place are applied in both places.
Port	From 0 to 65535	Enter the port of a SIP server that accepts registrations. The default is '0'.
Transport	DNSnaptr, UDPOnly, TCPpreferred, TCPOnly, TLS	Choose a transport method that the phone uses to communicate with the SIP server. There are four transport methods: DNSnaptr: The phone performs NAPTR and SRV look-ups that discover the transport, ports, and servers. DNSnaptr is the default transport method. UDPOnly: Only UDP is used. TCPpreferred: TCP is preferred and UDP is used if TCP fails. TCPOnly: Only TCP is used. TLS: If TLS is used, leave the port field empty, and the phone will use 5061 by default or you can set the port to 5061. If TLS fails, transport fails.
Expires (s)	10 to 2147483647	The phone's requested registration period. The value must be at least 10 seconds. The default is 3600 seconds.
Register	Yes, No	If set to No, calls can be routed to an outbound proxy without registration.
Retry Timeout (ms)	0 to 65535	This setting specifies how often retries will be sent. If you don't specify a value, or the value is 0, the standard RFC 3261 signaling retry behavior (the default behavior) is used. The minimum value is 100 milliseconds.
Retry Maximum Count	0 to 20	The number of retries to be attempted before moving on to the next available server. If you don't specify a value, or the value is 0, a value of 3 is used. The default is 3 retries.
Line Seize Timeout (s)	From 0 to 65536	The requested line-seize subscription period (from 0 to 65535 seconds). This is the amount of time to play the dial tone while the phone is off-hook before going back to the idle state. The default is 30 seconds.

This same information can be set for a second server which will be used if the primary server is not accessible. In addition, you can permanently forward calls to a different number with the Call Diversion options. See the Polycom UC Software Administrators Guide 4.0.1 for additional information.

An example configuration to register a line as extension 1029 to a call server at 172.22.2.203 will look like the figure shown next. Your settings will depend on the particular values required for your installation.

Registering aLine to the Call Server



Server1

Address	172.22.2.203
Port	0
Transport	DNSnaptr
Expires (s)	3600
Register	● Yes ○ No
Retry Timeout (ms)	0
Retry Maximum Count	3
Line Seize Timeout (s)	30

Using the SoundStructure VoIP Interface with SoundStructure Studio

As described in earlier sections, you can use SoundStructure Studio to configure the network settings of the SoundStructure VoIP Interface. You can then use either a provisioning server or the Web Configuration Utility to set up the VoIP-specific parameters.

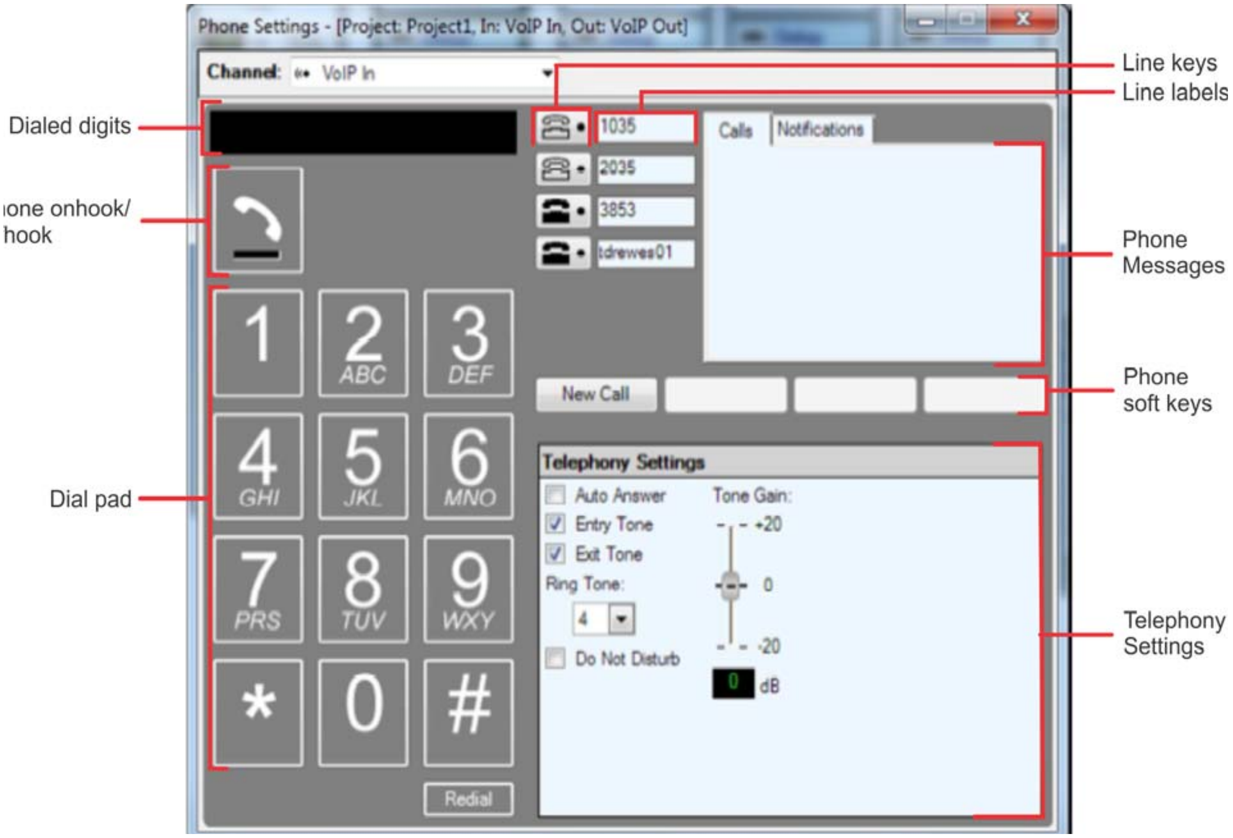
In addition to its use in setting up the system, SoundStructure Studio can also help with understanding how to control the SoundStructure VoIP Interface and to test the system by making audio calls.

Using the Phone Settings Control

You can use the phone settings control on the Channels Page in SoundStructure Studio to dial new calls, transfer calls, put calls on hold, join calls and split calls.

The user interface within SoundStructure Studio has been designed to look like the user interface of the SoundPoint IP phones with a dial keypad, line keys, and phone soft keys for initiating and managing calls as shown in the following figure. In addition, there are telephony settings for customizing auto-answer mode and entry and exit tones and there is a message waiting indicator that displays when there is a message for any of the line registrations.

Phone Settings in SoundStructure Studio


Phone Settings - [Project: Project1, In: VoIP In, Out: VoIP Out]

Channel: VoIP In

Line labels: 1035, 2035, 3853, tdrewes01

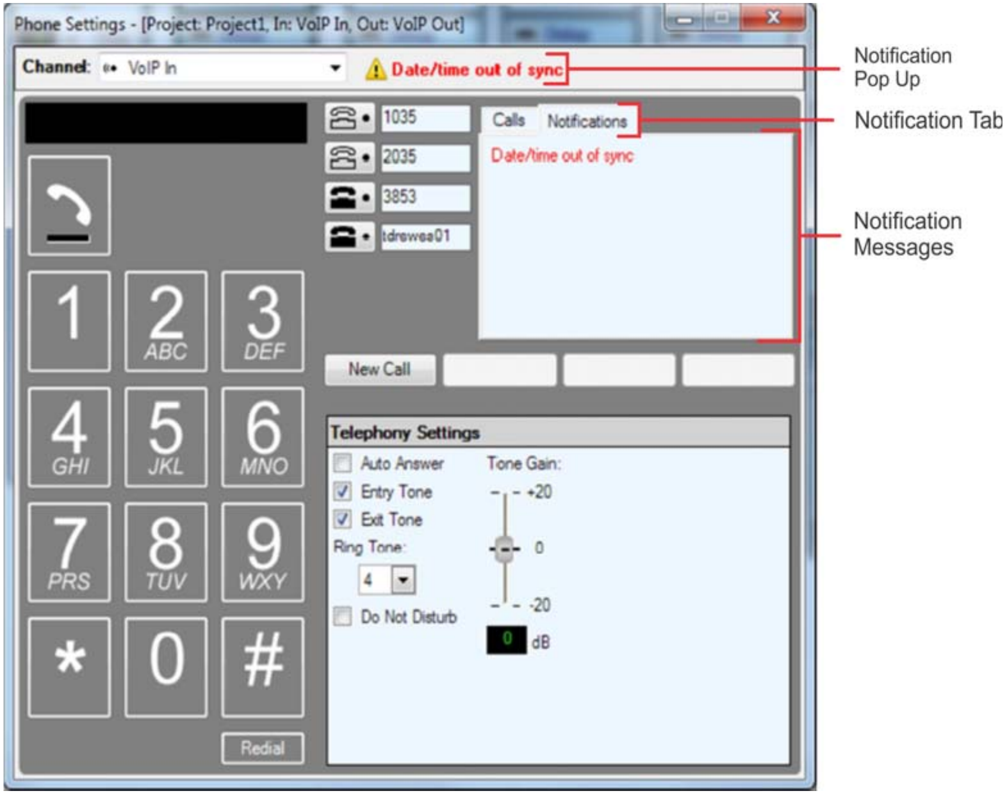
Telephony Settings

☐ Auto Answer
☑ Entry Tone
☑ Exit Tone
Ring Tone: 4
☐ Do Not Disturb
Tone Gain: 0 dB

Line keys
Line labels
Dialed digits
ione onhook/
hook
Phone
Messages
Calls
Notifications
New Call
Phone
soft keys
Dial pad
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PRS
8 TUV
9 WXY
*
0
#
Redial
+20
0
-20
Telephony
Settings

The following figure displays the Notifications tab, messages, and pop up that displays in Phone Settings.

Notifications in Phone Settings



Phone Settings - [Project: Project1, In: VoIP In, Out: VoIP Out]

Channel: VoIP In

Date/time out of sync

1035

2035

3853

tdrewes01

Date/time out of sync

Telephony Settings

☐ Auto Answer

☑ Entry Tone

☑ Exit Tone

Ring Tone: 4

☐ Do Not Disturb

Tone Gain: 0 dB

Notification Pop Up
Notification Tab
Notification Messages
Calls
Notifications
New Call
Redial
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PRS
8 TUV
9 WXY
*
0
#
+20
0
-20

A detailed description of the different UI elements is shown in the following table.

UI Elements

Control	Description
Line Keys	Shows all lines that have been defined with call servers. Up to 12 lines can be shown here. Lines 1-6 are shown on the left side of the Phone UI messages area and Lines 7 - 12 are shown in the right side of the Phone UI messages area. If the line is registered, the line icon will display as solid ☎ or if the line is not registered, the line icon will display as an outline ☏.
Line Labels	The label associated with the particular line.
Phone UI Messages	Shows any messages from the phone such as showing the calling party information. This area will behave similarly to the display in the SoundPoint IP phones.
Phone soft keys	The keys displayed vary depending on the call state and can be used to get a new line, end a call, put a call on hold, resume a call, transfer a call, and create conference calls.

UI Elements

Control	Description
Audio Settings	The audio settings allow an integrator to customize the in-room behavior of the SoundStructure VoIP Interface with auto answer, auto hangup, and ringtone selection.
Dial pad	The dial pad allows the integrator to dial digits to make a call to test the system.
Phone onhook/offhook	Indicates whether the call is offhook (`phone_connect` = 1) or onhook (`phone_connect` = 0).
Dialed digits	Displays the dialed digits for the current call.
Notifications Tab	Displays urgent messages about the VoIP interface's settings.

Customizing SoundStructure Telephony Settings

You can use the telephony settings to customize the behavior of the SoundStructure VoIP Interface with the following options.

Telephony Settings

Name	Possible Values	Description
Auto Answer	Enable or Disable (default)	Enables or disables answering the phone automatically after the second ring.
Entry Tone	Enable (default) or Disable	Determines whether the system plays a tone when the phone is auto answered.
Exit Tone	Enable (default) or Disable	Determines whether the system plays a tone when the phone auto hangs up after the remote caller hangs up.
Ring Tone	**1** Normal ring (default) **2** Low trill **3** Low double trill **4** Medium trill **5** Medium double trill **6** High trill **7** High double trill **8** Highest trill **9** Highest double trill **10** Beeble **11** Triplet **12** Ring splash **13** Low trill precedence **14** Silent	Customizes the default ringtone to one of 14 values. The default ringtone corresponds to the Default Ringtone in the Web Configuration Utility under **Preferences** > **Ringtones** > **Default Ringtone**. You can further customize the ringtone for an individual line from the Web Configuration Utility under **Settings** > **Lines** > **Line 1** > **Ring Type**. The default value for the Ring Type is to use the Default Ringtone setting from **Preferences** > **Ringtones**

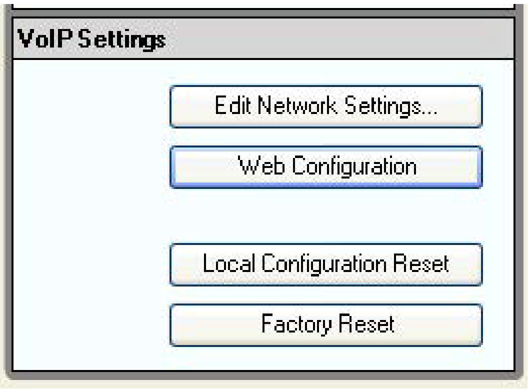
Telephony Settings

Name	Possible Values	Description
Tone Gain	-20 to +20 dB. Default is 0.	Sets the level of the entry and exit tones.
Do Not Disturb	Enable or Disable (default)	Determines whether the entire SoundStructure VoIP Interface is in Do Not Disturb mode. This setting will affect all line registrations and is not adjustable on a per line basis.

SoundStructure VoIP Interface Settings on the Wiring Page

The wiring page has several options for configuring the SoundStructure VoIP Interface that are active only when connected online with a SoundStructure system that has a SoundStructure VoIP Interface.

SoundStructure VoIP Interface Settings



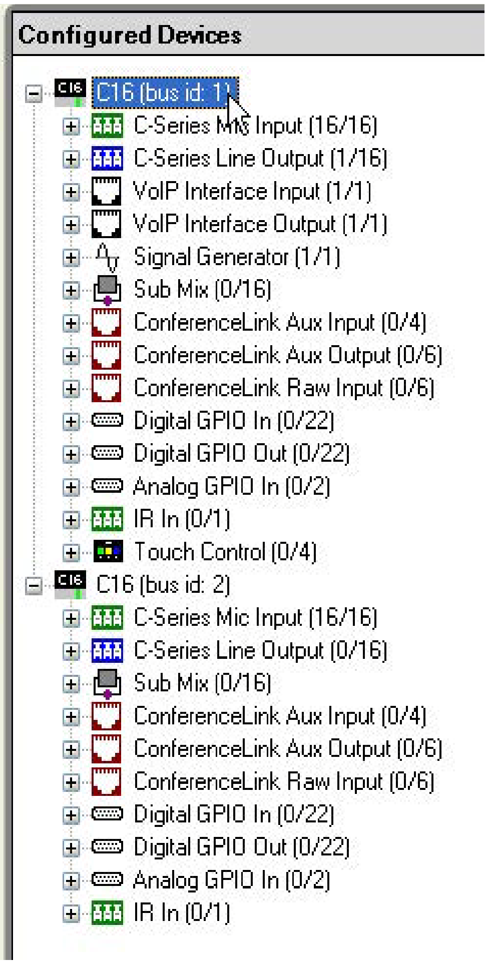
Wiring Page Options

Button Name	Description
Edit Network Settings	Opens the VoIP Network Settings user interface control for manually configuring the Ethernet settings for the SoundStructure VoIP Interface.
Web Configuration	Launches the default browser with the IP address of the SoundStructure VoIP Interface.
Local Configuration Reset	Resets to default values the auto answer, entry and exit tones, Do Not Disturb, and ringtone selection settings in the Telephony Settings. You will be prompted to the confirm after selecting this option.
Factory Reset	Resets the SoundStructure VoIP Interface to its factory default settings. This will clear all Ethernet settings, provisioning server settings, line registrations, and all other VoIP-specific parameters. You will be prompted to confirm after selecting this option. Before using this option you may want to use the Utilities > Phone Backup and Restore options from the Web Configuration Utility to save the VoIP-specific settings.

Setting an IP address with SoundStructure Studio

You can set the IP address and other Ethernet settings of the SoundStructure VoIP Interface via SoundStructure Studio when connected to an online SoundStructure system. To set Ethernet settings use the following steps:

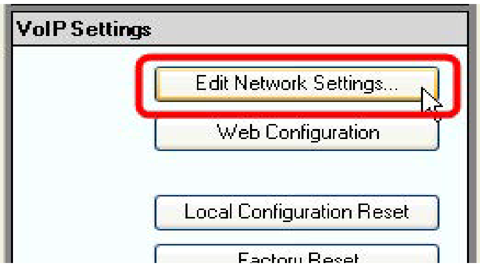
1. Connect to the SoundStructure system with SoundStructure Studio and select the Wiring page
2. Left click on the desired SoundStructure device with the SoundStructure VoIP Interface device as shown next. For SoundStructure system with only one device, the device is already selected.



Configured Devices

- C16 (bus id: 1)
  - C-Series Mic Input (16/16)
  - C-Series Line Output (1/16)
  - VoIP Interface Input (1/1)
  - VoIP Interface Output (1/1)
  - Signal Generator (1/1)
  - Sub Mix (0/16)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)
  - Touch Control (0/4)
- C16 (bus id: 2)
  - C-Series Mic Input (16/16)
  - C-Series Line Output (0/16)
  - Sub Mix (0/16)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

3. Next click on the Edit Devices Settings portion of the wiring page and then click **Edit Network Settings** as shown next.



Note: Working Online Only for SoundStructure VoIP Interface

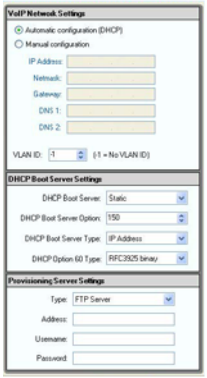
You can only set the Network Settings for the SoundStructure VoIP Interface when working online. The VoIP settings are grayed out when the system is offline or when the SoundStructure VoIP Interface is booting.

The Edit Network Settings dialog is divided into the following sections.

- VoIP Network Settings to set the Ethernet parameters.
- DHCP Boot Server Settings to determine how much information the SoundStructure VoIP Interface will get from the DHCP server.
- Provisioning Server Settings to manually configure the provisioning server settings if the DHCP Boot Server settings are set to Static or the Ethernet settings are set to Manual.

These areas display as shown in the following figure.

Edit Network Settings Areas


VoIP Network Settings
IP Address:
Netmask:
Gateway:
DNS 1:
DNS 2:
DHCP Boot Server Settings
Provisioning Server Settings
Address:
Username:
Password:

● Automatic configuration (DHCP)
○ Manual configuration

VLAN ID: -1 (-1 = No VLAN ID)

DHCP Boot Server: Static
DHCP Boot Server Option: 150
DHCP Boot Server Type: IP Address
DHCP Option 60 Type: RFC3925 binary

Type: FTP Server

VoIP Network Settings

Select either Automatic Configuration (DHCP) or Manual Configuration for configuring the IP address of the SoundStructure VoIP Interface as described in the following table.

IP Address Settings

Name	Possible Values	Description
Address Mode	Automatic Configuration (DHCP) (default)	The SoundStructure VoIP Interface expects to receive an IP address from the network DHCP server. If you select automatic configuration then the manual fields are grayed out and are not accessible.
	Manual configuration	You will need to manually set the SoundStructure VoIP Interface’s IP address including the IP address, netmask, gateway, and optional DNS servers.

If you select the Manual configuration option, you must enter the information for the following fields to properly configure the IP address of the SoundStructure VoIP Interface.

IP Address Settings

Name	Possible Values	Description
IP Address	Dotted-decimal IP address	Enter the IP address to be used for the SoundStructure VoIP Interface. This IP address should be unique to the SoundStructure VoIP Interface.
Netmask	Dotted-decimal subnet mask	Enter the subnet netmask to be used with the IP address. Typically this is equal to 255.255.255.0.
Gateway	Dotted-decimal IP address	Enter the IP address of the router that is the address the SoundStructure VoIP Interface will go to when seeking IP addresses outside of the local subnet.
DNS 1	Dotted-decimal IP address	Enter the IP address of the primary domain name server.
DNS 2	Dotted-decimal IP address	Enter the IP address of the secondary domain name server.
VLAN ID	-1, or 0 to 4095	Set the VLAN ID to -1 to indicate no VLAN ID, or to the value of the desired VLAN ID.

DHCP Boot Server Settings

If you are using DHCP to set the IP address of the SoundStructure VoIP Interface, then the DHCP server can also supply additional settings to the SoundStructure VoIP Interface to simplify setup and configuration. Enter the parameters described in section Setting the Provisioning Server Information.

Provisioning Server Settings

If you are using a central provisioning server or temporarily setting up a manual FTP server for a firmware update, you may enter the server information on this page including the type of access to the provisioning server, the server address, username, and password information as shown in the following table.

Provisioning Settings

Name	Possible Values	Description
Type	FTP, TFTP, HTTP, HTTPS, FTPS	The protocol that the SoundStructure VoIP Interface will use to obtain configuration and phone application files from the provisioning server.
Address	dotted-decimal IP address OR domain name string OR URL All addresses can be followed by an optional directory and optional filename.	The provisioning server to use if the DHCP client is disabled, the DHCP server does not send a boot server option, or the Boot Server parameter is set to Static. The SoundStructure VoIP Interface can contact multiple IP addresses per DNS name. These redundant provisioning servers must all use the same protocol. If a URL is used it can include a user name and password. **Note:** ":", "@", or "/" characters can be used in the user name or password these if they are correctly escaped using the method specified in RFC 1738.
Username	any string	The user name used when the SoundStructure VoIP Interface logs in to the server if required for the selected server Type
Password	any string	The password used when the SoundStructure VoIP Interface logs into the server if required for the selected server Type. **Note:** If the Server Address is a URL that includes the user name and password, the password field will be ignored.

Changes to the network settings for the *SoundStructure VoIP Interface* are stored into the permanent memory of the SoundStructure VoIP Interface immediately when you press the **Apply** button. Note that this behavior is different from how changes are made to the *SoundStructure device's* network settings. SoundStructure device network changes require that you save the project before the network setting changes are stored permanently into the SoundStructure device.

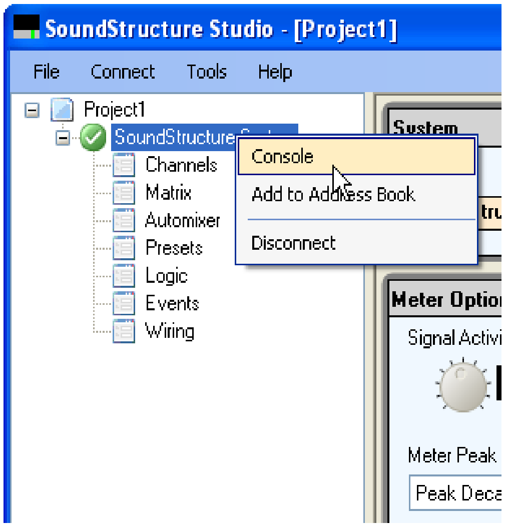
Changing the network settings may cause the SoundStructure VoIP Interface to reboot. When this happens, you may hear a series of tones played out the local loudspeaker system indicating that the SoundStructure VoIP Interface has initiated a reboot. If a reboot happens, the wiring page in SoundStructure Studio will update the status of the SoundStructure VoIP Interface to 'booting' and the rear panel status LED will blink until the device has finished booting and then turn solid once the card has finished booting.

Using the SoundStructure Studio Console

You can use the SoundStructure Studio console to better understand the API commands that are used to control the SoundStructure system for dialing, transferring, putting calls on hold, and more. You can open the SoundStructure Studio console window to see the SoundStructure API commands that are sent to the SoundStructure device while configuring the system, and follow the acknowledgment that are returned. Please note that you need to be connected online to control a SoundStructure VoIP Interface.

To open the console, right-click on the project name and select **Console** as shown next.

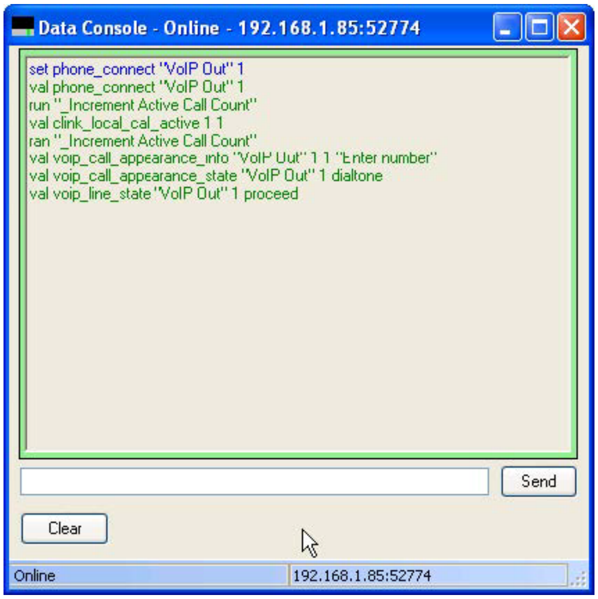
Console in SoundStructure Studio


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring
Console
Add to Address Book
Disconnect
System
Meter Optio
Signal Activi
Meter Peak
Peak Deca

Once the console is open you can see the commands associated with the user interface controls within SoundStructure Studio.

For example, taking the phone offhook from the channels page will result in the `phone_connect` command being sent (in blue) and the acknowledgment from the system (in green) returned as shown in the following figure. This example shows other acknowledgments in addition to the `phone_connect` acknowledgment. Depending on your system configuration and programmed events, you may other command acknowledgments as a consequence of taking the phone offhook.

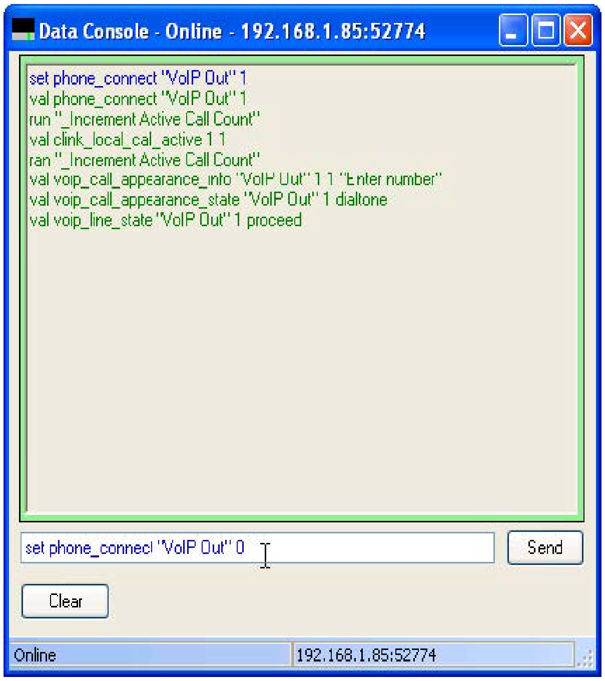
Data Console Acknowledgments


Data Console - Online - 192.168.1.85:52774
Send
Clear
Online
192.168.1.85:52774

```
set phone_connect "VoIP Out" 1
val phone_connect "VoIP Out" 1
run "_Increment Active Call Count"
val clink_local_cal_active 1 1
ran "_Increment Active Call Count"
val voip_call_appearance_info "VoIP Out" 1 1 "Enter number"
val voip_call_appearance_state "VoIP Out" 1 dialtone
val voip_line_state "VoIP Out" 1 proceed
```

You can enter commands by typing text in the white area and pressing Enter or the Send button as shown in the following figure.

Entering Commands in the Data Console



```
set phone_connect "VoIP Out" 1
val phone_connect "VoIP Out" 1
run "_Increment Active Call Count"
val clink_local_cal_active 1 1
ran "_Increment Active Call Count"
val voip_call_appearance_info "VoIP Out" 1 1 "Enter number"
val voip_call_appearance_state "VoIP Out" 1 dialtone
val voip_line_state "VoIP Out" 1 proceed
```

```
set phone_connect "VoIP Out" 0
```

Updating Software on the SoundStructure VoIP Interface

There are three types of software associated with the SoundStructure VoIP Interface:

- SoundStructure Studio software for Windows PC's
- SoundStructure device firmware
- SoundStructure VoIP Interface firmware

In this section, you will learn how to upgrade the software in the SoundStructure VoIP Interface. SoundStructure device firmware upgrades were described in Upgrading the Firmware in the SoundStructure System.

In the event there is new software available from Polycom's Web site for SoundStructure VoIP Interface, you can download that firmware and use it to update the plug-in card. You can update the software of the SoundStructure VoIP Interface in several ways, including using a local PC-based FTP/HTTP/TFTP server or using a central provisioning server via FTP or HTTP.

Upgrading Software with a Local FTP Server

A simple way to update software of the SoundStructure VoIP Interface is to use a File Transfer Protocol or FTP program. Although FTP servers are free, they require installation, and use logins and passwords.

Note: Turning off Windows Firewall to Run a FTP Server

To run the FTP server on your local PC, you will either have to temporarily turn **off** your Windows Firewall or open port 21 to allow FTP traffic to your PC. If you do not allow traffic through your Windows Firewall, your SoundStructure VoIP Interface will not be able to reach your PC's FTP server and you will not be able to update the software on your SoundStructure VoIP Interface. After the software has been updated, you can turn your firewall back on or close any ports you opened.

Downloading and installing an FTP server

A free and popular server, Filezilla Server, is available for Windows at http://filezilla-project.org. This application (version 0.9.xx) has been tested with the UC Software.

To set up the FTP server:

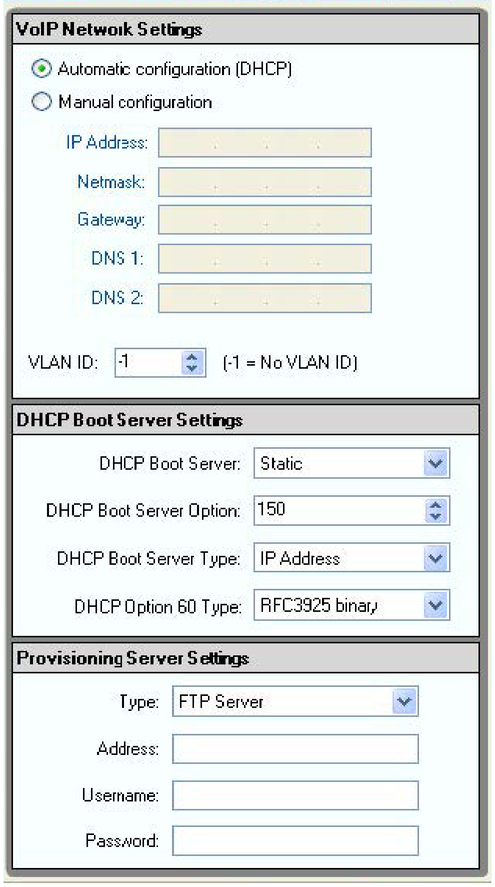
1. Download and install the latest version of Filezilla Server. For example, visit http://filezilla-project.org/.
2. After installation, a 'Connect to Server' dialog with display. Select **OK** to open the administrative user interface.
3. To configure a user, select **Edit** > **Users** in the status bar.
4. Select **Add.**
5. Enter the user name for the phone, for example, **bill123**, and select **OK**.
6. Select the **Password** check box and enter a password, for example, **1234.** The phone will use this password to log in.
7. Select **Page** >**Shared folders** to specify the server-side directory where the provisioning files and/or software files will be located (and the VoIP log files uploaded).
8. Select **Add** and pick the directory.
9. To allow the phone to upload logs onto the provisioning server, select the **Shared Folders** > **Files** >**Select Write and Delete** check boxes, and then select **Shared Folders** > **Files** >**Select Write and Delete** check boxes, and then select **OK.**
10. Determine the IP address of the FTP server by entering **cmd** in the Run dialog on your Start menu, and **ipconfig** in the command prompt. The resulting text shows the IP Address of the FTP server.

Follow the steps in the next section to upgrade the software on the SoundStructure VoIP Interface once the FTP server has been created. Use the username and password created above as the username and password for the provisioning server settings.

Upgrading Software with an Existing Provisioning Server

To update the SoundStructure VoIP Interface software via an existing provisioning server, follow these steps.

1 In the SoundStructure VoIP Interface Network settings on the Wiring Page, set the DHCP boot server to **Static** and set the Provisioning Server Settings to use an **FTP Server** and enter the appropriate server address, username, and password.



VoIP Network Settings

● Automatic configuration (DHCP)
○ Manual configuration
IP Address:
Netmask:
Gateway:
DNS 1:
DNS 2:
VLAN ID: -1 (-1 = No VLAN ID)

DHCP Boot Server Settings

DHCP Boot Server: Static
DHCP Boot Server Option: 150
DHCP Boot Server Type: IP Address
DHCP Option 60 Type: RFC3925 binary

Provisioning Server Settings

Type: FTP Server
Address:
Username:
Password:

2 Copy the new software file to the directory specified in step 7 of installing the FTP server. The software file will have a name of the form 3111-33215-001.sip.ld where 3111-33215-001 is the part number associated with the SoundStructure VoIP Interface software.

3 Copy the file 000000000000.cfg file to the directory specified in step 7 of installing the FTP server. This configuration file tells the SoundStructure VoIP Interface which firmware file to look for when it connects to the FTP server. The contents of this configuration file must have at least the information

shown below for the SoundStructure VoIP Interface to find and upload the desired sip.ld software file. The SoundStructure VoIP Interface will know to look for a file named 3111-33215-001.sip.ld or sip.ld based on the file contents shown next.

```
<APPLICATION APP_FILE_PATH="sip.ld" CONFIG_FILES="" MISC_FILES=""
LOG_FILE_DIRECTORY="" OVERRIDES_DIRECTORY="" CONTACTS_DIRECTORY=""
LICENSE_DIRECTORY="" USER_PROFILES_DIRECTORY="" CALL_LISTS_DIRECTORY="">
```

To select a particular version of software to load to a particular SoundStructure VoIP Interface, create a configuration file with a name <MAC Address>.cfg where <MAC Address> is the MAC address of the SoundStructure VoIP Interface (for example, 0004F2BF001D.cfg). In this file, specify the exact name of the software file to load with file contents of:

```
<APPLICATION APP_FILE_PATH="my_version_sip.ld"
CONFIG_FILES="sip-basic.cfg" MISC_FILES="" LOG_FILE_DIRECTORY=""
OVERRIDES_DIRECTORY="" CONTACTS_DIRECTORY="" LICENSE_DIRECTORY="">
</APPLICATION>
```

where my_version_sip.ld is the name of the target software filename.

4. Reboot the SoundStructure VoIP Interface to force the system to check the FTP site and load new software. The boot up process will take longer than usual as the software is loaded and installed.

   The SoundStructure VoIP Interface may be rebooted either by the Web Configuration utility under Utilities > Reboot phone or from the SoundStructure Studio Console by typing the command

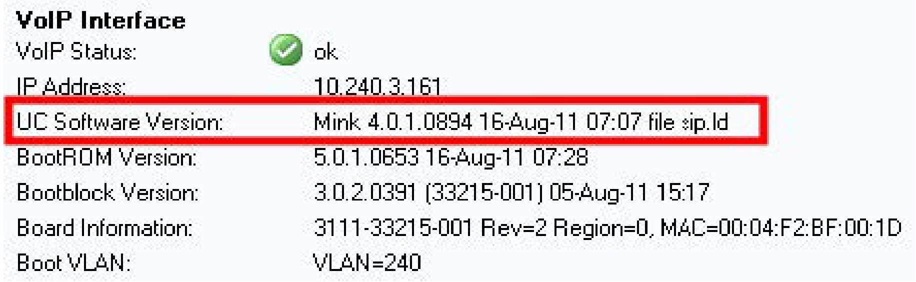
   ```
   set voip_reboot “VoIP Out”
   ```

   which will cause the voip_reboot acknowledgment to be sent from the SoundStructure system:

   ```
   val voip_reboot “VoIP Out”
   ```

   where “VoIP Out” is the name of the output channel of the interface to reboot. If you have named your channel something else, use the name you have selected.

5. Once the SoundStructure VoIP Interface has finished booting (solid green light on the VoIP status LED), check that the software version has been on updated by confirming the version number on the SoundStructure Studio wiring as shown below.



VoIP Interface

VoIP Status:	✓ ok
IP Address:	10.240.3.161
UC Software Version:	Mink 4.0.1.0894 16-Aug-11 07:07 file sip.ld
BootROM Version:	5.0.1.0653 16-Aug-11 07:28
Bootblock Version:	3.0.2.0391 (33215-001) 05-Aug-11 15:17
Board Information:	3111-33215-001 Rev=2 Region=0, MAC=00:04:F2:BF:00:1D
Boot VLAN:	VLAN=240

Upgrading Software with the Web Configuration Utility

The Web Configuration Utility provides a convenient way to manage and upgrade the software in your SoundStructure VoIP Interface through a local FTP server or local HTTP server.

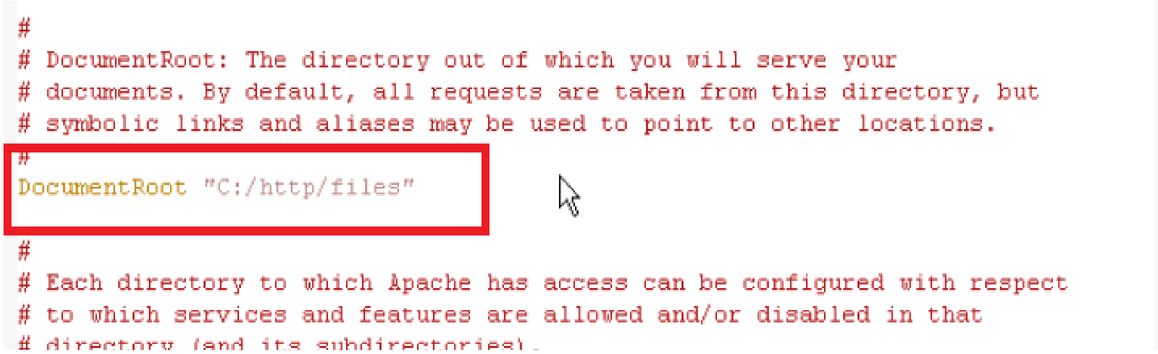
Note: Turning off Windows Firewall to Run a FTP or HTTP Server

To run either an FTP or HTTP server on your local PC, you will have to temporarily turn **off** or open the appropriate ports on your Windows Firewall. If you do not allow traffic through your Windows Firewall, or make exceptions to allow the FTP or HTTP traffic, your SoundStructure VoIP Interface will not be able to reach your PC's FTP or HTTP server and you will not be able to update the software on your SoundStructure VoIP Interface. After the software has been updated, you can turn your firewall back on or close any ports you opened.

Downloading and Installing an HTTP Web Server

A free and popular Web server, Apache Server, is available for 32-bit Windows at http://httpd.apache.org/download.cgi#apache22 and version 2.2.19 has been tested with the Polycom UC Software. To set up the Web server:

1. Download and install the latest version of Apache Server from http://httpd.apache.org/download.cgi#apache22.
2. After installation test that the web server works by opening your browser and entering in the IP address of your PC. If successful you should see a page with the text "It works".
3. Edit the httpd.conf file that by default will be installed in the c:\Program Files\Apache Software Foundation\Apache2.2\conf directory and set the DocumentRoot entry to point to the directory where you would like to have the software files, for example enter **c:/http/files** for the Windows directory of c:\http\files.



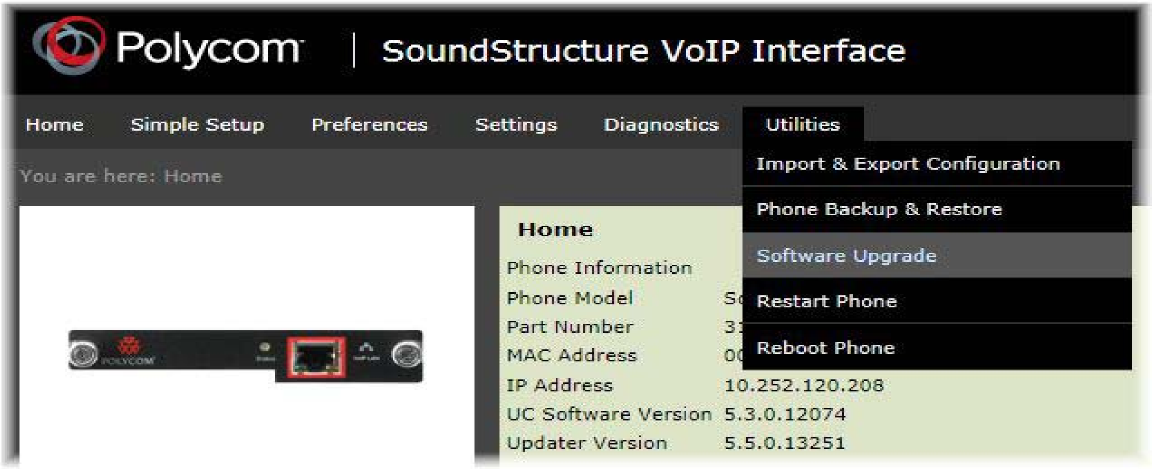
```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:/http/files"

#
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
```

Using a Web Server with Web Configuration Utility Software Upgrade

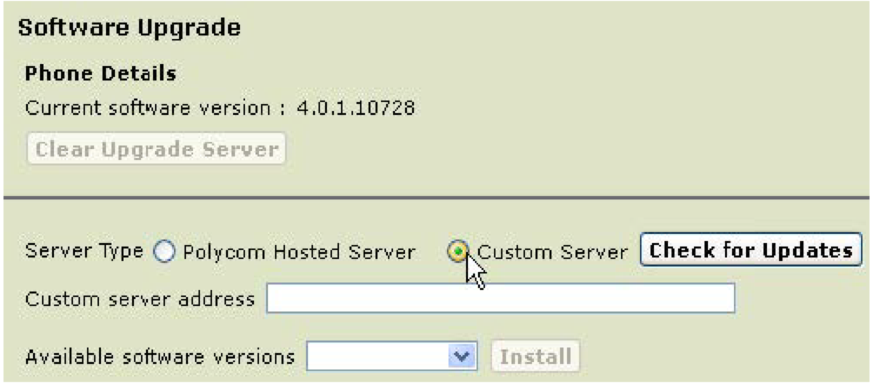
To use the Web Configuration Utility Software Upgrade feature, follow these steps:

1 Within the Web Configuration Utility navigate to the **Utilities** > **Software Upgrade** menu of the Web Configuration Utility as shown in the following figure.


Polycom
SoundStructure VoIP Interface
Home
Simple Setup
Preferences
Settings
Diagnostics
Utilities
Import & Export Configuration
Phone Backup & Restore
Software Upgrade
Restart Phone
Reboot Phone
You are here: Home
Home

Phone Information
Phone Model Sc
Part Number 31
MAC Address 00
IP Address 10.252.120.208
UC Software Version 5.3.0.12074
Updater Version 5.5.0.13251

2 Select the **Custom Server** option as shown in the following figure.


Software Upgrade
Phone Details
Clear Upgrade Server
Check for Updates
Install

Current software version : 4.0.1.10728

Server Type ○ Polycom Hosted Server ● Custom Server

Custom server address

Available software versions

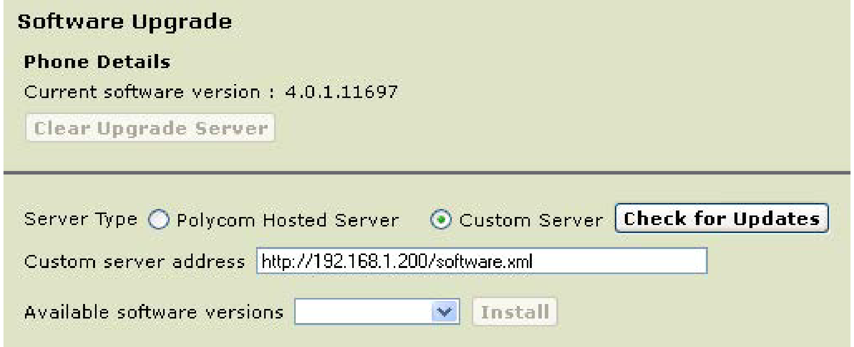
3 Create an XML file on the web server (in the c:\http\files directory in this example) in the format shown below with at least one Phone_Image entry:

```
<PHONE_IMAGES>
 <REVISION id="3111-33215-001">
  <PHONE_IMAGE>
    <VERSION>4.0.1.7555</VERSION>
    <PATH> http://192.168.1.200/UCS_4.0.1_rts30/ </PATH>
  </PHONE_IMAGE>
  <PHONE_IMAGE>
    <VERSION>4.0.1.10728</VERSION>
    <PATH> http://192.168.1.200/UCS_4.0.1_rts43/ </PATH>
  </PHONE_IMAGE>
  <PHONE_IMAGE>
    <VERSION>4.0.1.11697</VERSION>
    <PATH> http://192.168.1.200/UCS_4.0.1_rts46/ </PATH>
  </PHONE_IMAGE>
 </REVISION>
</PHONE_IMAGES>
```

where the names UCS_4.0.1_rts30, UCS_4.0.1_rts43, and UCS_4.0.1_rts46 are directory names that store their respective versions of the 3111-33215-001.sip.ld software files. These directories must match the path specified in the XML file and the directories should be located in the main directory that was configured for the web server. The contents of these subdirectories must include the file 3111-33215-001.sip.ld that corresponds to the version specified in the software.xml file.
In this example, the web server at 192.168.1.200 has a root directory that was defined during the installation of the web server to be c:\http\files. The directories UCS_4.0.1_rts30, etc., are subdirectories of this root directory. In this example these directories are subdirectories of the directory c:\http\files\.

4 Enter the IP address of a web server along with the name of the xml file that contains the appropriate firmware versions created in the previous step in the **Custom server address** field.

For example: http://192.168.1.200/software.xml
as shown in the following figure.



Software Upgrade

Phone Details

Current software version : 4.0.1.11697

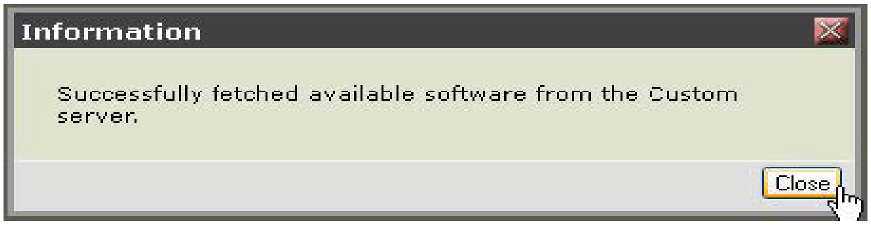
Server Type ○ Polycom Hosted Server ● Custom Server

Custom server address http://192.168.1.200/software.xml

Available software versions

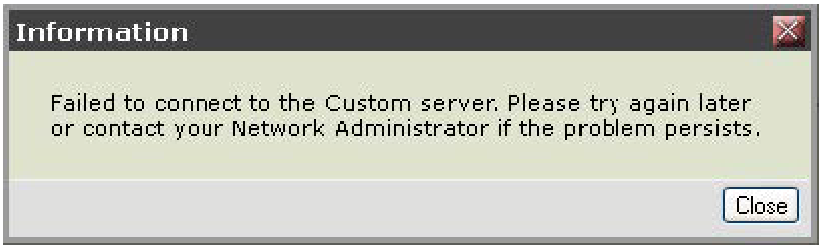
Note that steps 3 and 4 could also use an FTP server by replacing all the references to *http://* with *ftp://username:password@* in the example XML file in step 3 and in the address of the server in this step. For example, you could use ftp://bill123:1234@192.168.1.200/software.xml instead of http://192.168.1.200/software.xml, where bill123 was the username and 1234 the password configured for the ftp server.

5 From the Web Configuration Utility, select Check for Updates. If the web server can be reached and new software versions found, the Web Configuration Utility will indicate that new versions of software were found as shown in the following figure.



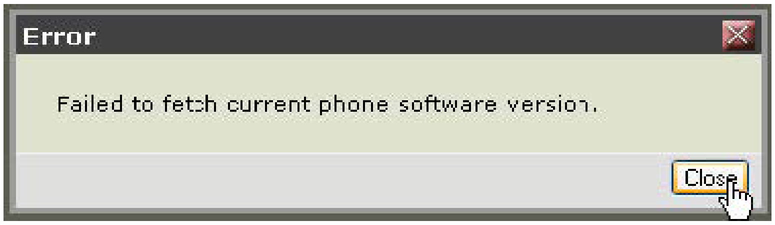
Successfully fetched available software from the Custom server.

If the web server cannot be found then the system will present an error message of the form shown in the following figure. If you see this message, check that the SoundStructure VoIP Interface and the HTTP or FTP server have a valid network route and that your Windows Firewall is disabled if you are using your local PC’s FTP server or HTTP server.



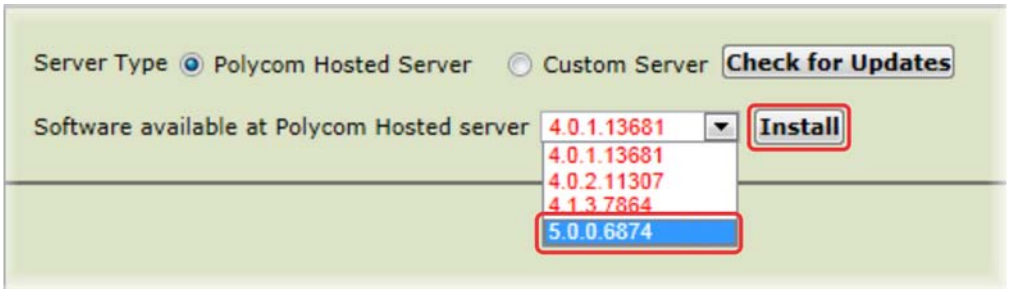
Failed to connect to the Custom server. Please try again later or contact your Network Administrator if the problem persists.

If the specified XML file is not found, then an error message will display indicating that it is not possible to communicate to the phone as shown next.



Failed to fetch current phone software version.

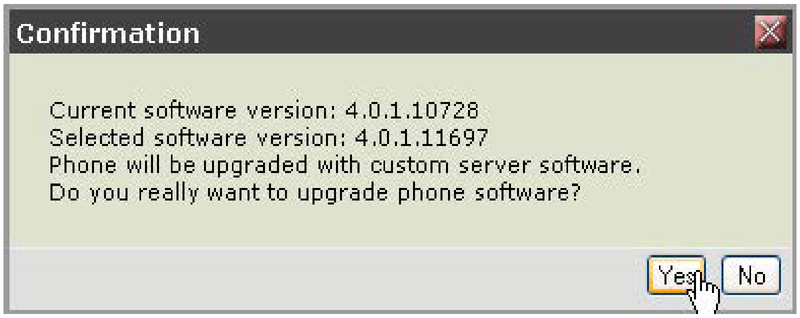
If software versions are found, select the version of software from the Software menu and select **Install** as shown in the following figure.



Server Type ● Polycom Hosted Server ○ Custom Server

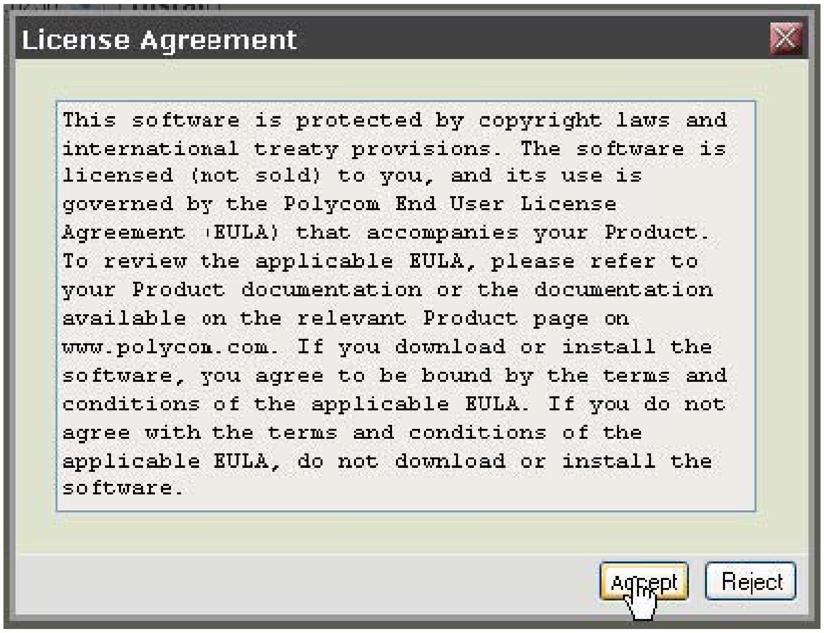
Software available at Polycom Hosted server 4.0.1.13681

6 Versions of firmware that are newer than the current running software are colored green, the current software version is colored blue, and previous versions of software are colored red. After clicking Install, you will be presented with a confirmation to proceed. Select **Yes** to continue or No to cancel the software upgrade.



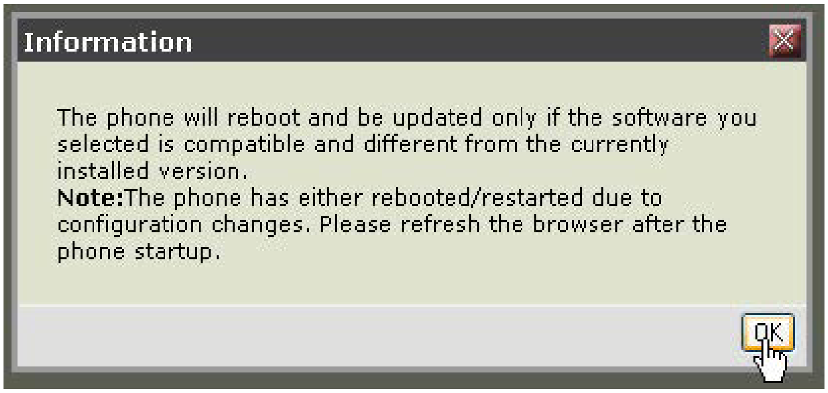
Current software version: 4.0.1.10728
Selected software version: 4.0.1.11697
Phone will be upgraded with custom server software.
Do you really want to upgrade phone software?

7 If you select Yes, then you will be presented with a license agreement to accept or reject. To continue, select Accept.



This software is protected by copyright laws and international treaty provisions. The software is licensed (not sold) to you, and its use is governed by the Polycom End User License Agreement (EULA) that accompanies your Product. To review the applicable EULA, please refer to your Product documentation or the documentation available on the relevant Product page on www.polycom.com. If you download or install the software, you agree to be bound by the terms and conditions of the applicable EULA. If you do not agree with the terms and conditions of the applicable EULA, do not download or install the software.

8 Finally, the system will begin the upgrade process by rebooting the phone and beginning the firmware file transfer. Press Ok to continue.



Information

The phone will reboot and be updated only if the software you selected is compatible and different from the currently installed version.
Note: The phone has either rebooted/restarted due to configuration changes. Please refresh the browser after the phone startup.

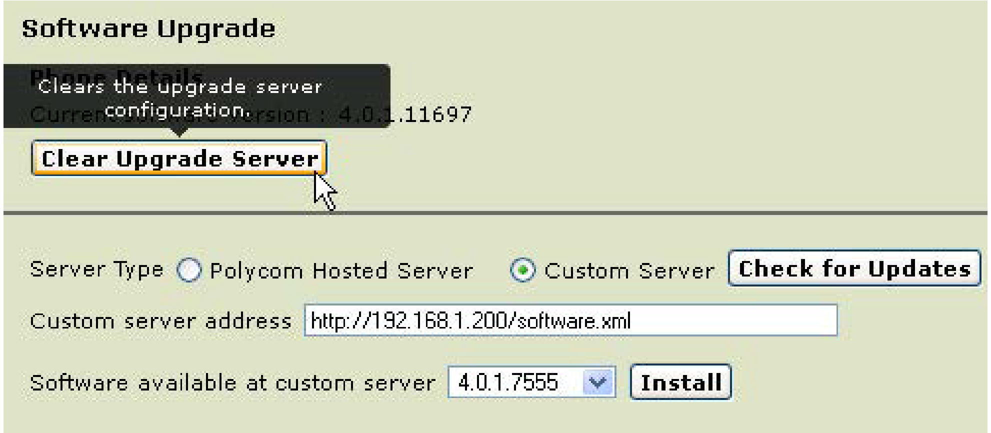
9 Once the system has completed, confirm the SoundStructure VoIP Interface has the desired version of firmware from the Wiring page.

Note: Clearing Software Upgrade Server to Complete Provisioning Server Firmware Update

After the software has been updated via the Web Configuration Utility, the standard provisioning server firmware update process will not work until the Software Upgrade Server has been cleared by clicking **Clear Upgrade Server** or the SoundStructure VoIP Interface has been factory reset.

To clear the Software Upgrade Server

1 Click the **Clear Upgrade Server** as shown in the following figure.



Software Upgrade

Phone Details

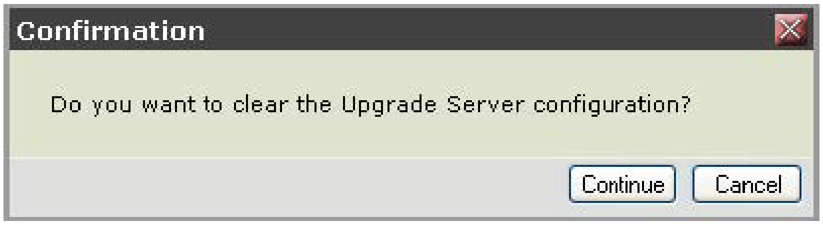
Current Software Version : 4.0.1.11697

Server Type ○ Polycom Hosted Server ● Custom Server

Custom server address http://192.168.1.200/software.xml

Software available at custom server 4.0.1.7555

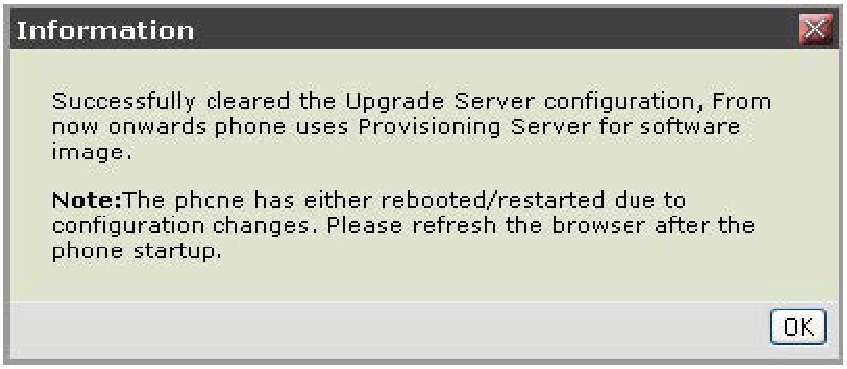
2 Click **Continue** when prompted.



Confirmation

Do you want to clear the Upgrade Server configuration?

A confirmation explaining that the server has been cleared. displays



Information

Successfully cleared the Upgrade Server configuration, From now onwards phone uses Provisioning Server for software image.

Note: The phcne has either rebooted/restarted due to configuration changes. Please refresh the browser after the phone startup.

Validating a SoundStructure VoIP Interface Installation

Once you have completed your SoundStructure configuration file, have uploaded the configuration file to the SoundStructure system, and configured the SoundStructure VoIP Interface, you can validate your installation with the following steps:

1. Confirm Version 1.7.0 or newer of SoundStructure Studio is used.
2. Confirm Version 1.5.0 or newer SoundStructure firmware is used.
3. Confirm a SoundStructure VoIP Interface channel is defined in the project.
4. Confirm the VoIP Status on Wiring page is ‘ok’.
5. Confirm the SoundStructure VoIP Interface has a valid IP address.
6. Confirm there is a valid network route to the SoundStructure VoIP Interface.
7. Confirm the line is registered - or dial a SIP URL call to confirm the SoundStructure VoIP Interface card has been installed properly except for line registration settings.
8. Confirm a call can be dialed using the registered line.

These steps are described in more detail in the following sections.

Version 1.7.0 or Newer of SoundStructure Studio Used

Version 1.7.0 SoundStructure Studio software is required for connecting to a SoundStructure system that has firmware version 1.5.0 or newer and has a SoundStructure VoIP Interface card installed.

Note: SoundStructure VoIP Interface Parameters Not Supported in Older Firmware Versions

Older versions of SoundStructure Studio do not support the latest parameter definitions required by the SoundStructure VoIP Interface and do not connect properly to a SoundStructure System that has a SoundStructure VoIP Interface installed. Older versions of SoundStructure Studio may exit prematurely with an error message about undefined parameters when trying to access online systems that use the SoundStructure VoIP Interface.

Use SoundStructure Studio version 1.7.0 or later to resolve this issue.

SoundStructure Device Firmware 1.5.0 or newer

SoundStructure device firmware older than 1.5.0 (e.g., version 1.3.3 and earlier) do not support the SoundStructure VoIP Interface. If a SoundStructure VoIP Interface is installed in a SoundStructure device with device firmware earlier than 1.5.0, the status LED on the SoundStructure VoIP Interface will not light up. To fix this problem, the firmware to version 1.5.0 or newer as described in Upgrading the Firmware in the SoundStructure System.

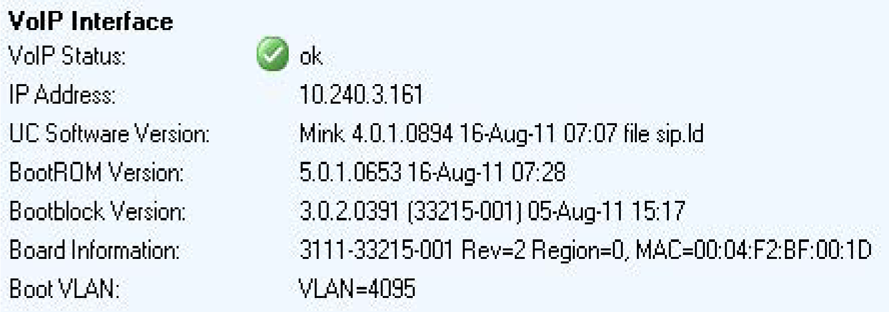
SoundStructure VoIP Interface Channel Defined in the Project

If the SoundStructure VoIP Interface is plugged into the SoundStructure device but no VoIP interface virtual channels are defined in the project, then you will see that VoIP settings are not present in the Wiring page in SoundStructure Studio. To fix this, click **Edit Channels** option and add a VoIP interface to the project.

VoIP Status is OK

Check that the VoIP status is set to **ok** on the wiring page with an online project. “Ok” means the SoundStructure VoIP Interface has finished booting, is communicating with the SoundStructure system, and is ready to be configured.

SoundStructure VoIP Status in SoundStructure Studio

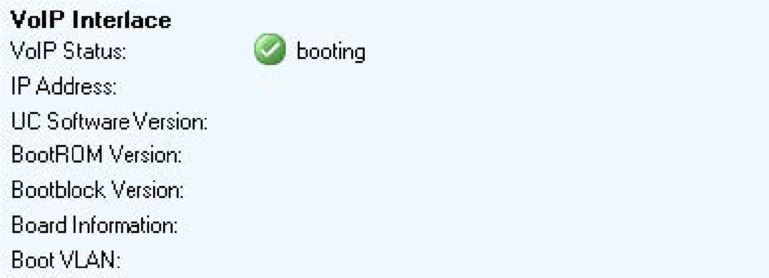


VoIP Interface

VoIP Status: ✓ ok
IP Address: 10.240.3.161
UC Software Version: Mink 4.0.1.0894 16-Aug-11 07:07 file sip.ld
BootROM Version: 5.0.1.0653 16-Aug-11 07:28
Bootblock Version: 3.0.2.0391 (33215-001) 05-Aug-11 15:17
Board Information: 3111-33215-001 Rev=2 Region=0, MAC=00:04:F2:BF:00:1D
Boot VLAN: VLAN=4095

If the SoundStructure VoIP Interface is booting, the VoIP status will indicate the device is booting and this section of the wiring page will appear as shown in the following figure. The SoundStructure VoIP Interface requires approximately 1 to 2 minutes to boot. When you perform a software update, the boot time will increase.

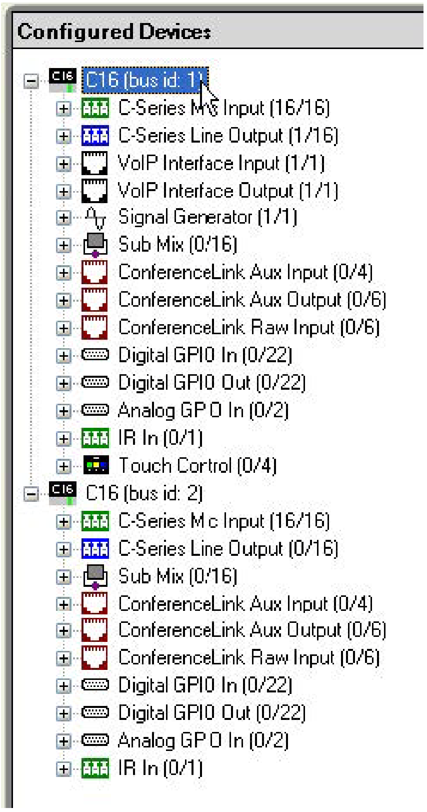
SoundStructure VoIP Interface Booting Status



VoIP Interface

VoIP Status:	✓ booting
IP Address:	
UC Software Version:	
BootROM Version:	
Bootblock Version:	
Board Information:	
Boot VLAN:	

If you don’t have a VoIP interface listed on your wiring page, even though you have a SoundStructure VoIP Interface plugged in, ensure that your project includes a SoundStructure VoIP Interface as described in the previous section. Check the Wiring page Configured Devices area to confirm you have a SoundStructure VoIP Interface installed into one or more of your SoundStructure devices as shown in the Configured Devices section shown next. If a SoundStructure VoIP Interface is installed in a multi-SoundStructure device system, first select the SoundStructure devices that has the SoundStructure VoIP Interface installed to see the VoIP interface info in the device information area.



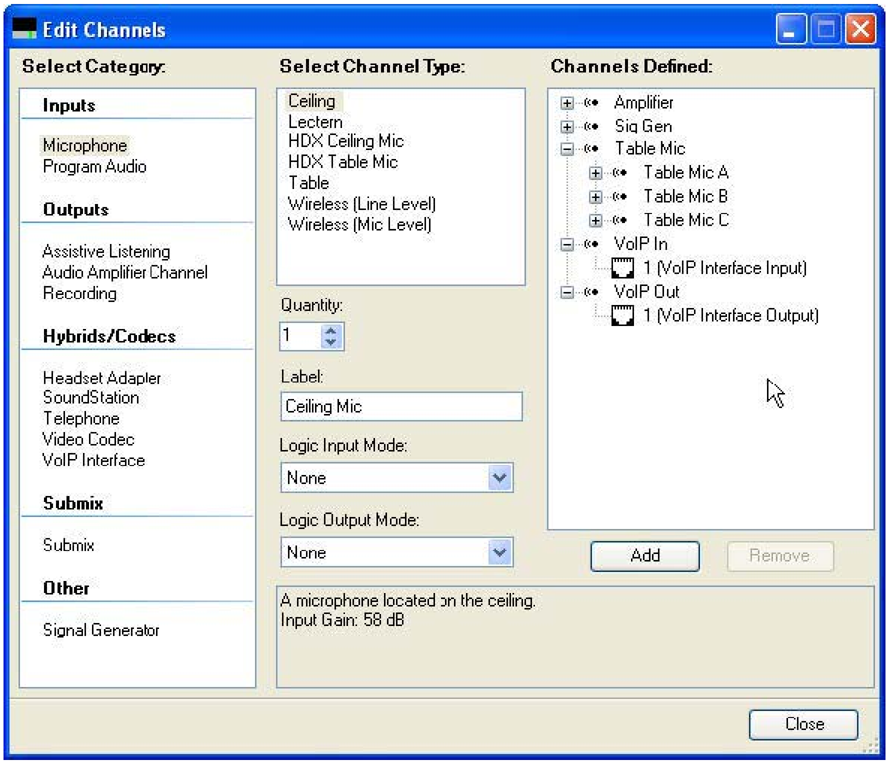
Configured Devices

- C16 (bus id: 1)
  - C-Series Mic Input (16/16)
  - C-Series Line Output (1/16)
  - VoIP Interface Input (1/1)
  - VoIP Interface Output (1/1)
  - Signal Generator (1/1)
  - Sub Mix (0/16)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)
  - Touch Control (0/4)
- C16 (bus id: 2)
  - C-Series Mic Input (16/16)
  - C-Series Line Output (0/16)
  - Sub Mix (0/16)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

After you’ve confirmed a SoundStructure VoIP Interface is part of the configured devices installed, check the **Edit Channels** control to ensure you have a SoundStructure VoIP Interface designed into your project.

While the channel names may be different in your project, in the following figure, the virtual channels “VoIP In” and “VoIP Out” are seen as defined in the system.

Project Names in Edit Devices Dialog


Edit Channels
Select Category:
Inputs
Microphone
Program Audio
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Headset Adapter
SoundStation
Telephone
Video Codec
VoIP Interface
Submix
Submix
Other
Signal Generator
Select Channel Type:
Ceiling
Lectern
HDX Ceiling Mic
HDX Table Mic
Table
Wireless (Line Level)
Wireless (Mic Level)
Channels Defined:
Add
Remove
Close

Channels Defined:

- Amplifier
- Sig Gen
- Table Mic
  - Table Mic A
  - Table Mic B
  - Table Mic C
- VoIP In
  - 1 (VoIP Interface Input)
- VoIP Out
  - 1 (VoIP Interface Output)

Quantity: 1

Label: Ceiling Mic

Logic Input Mode: None

Logic Output Mode: None

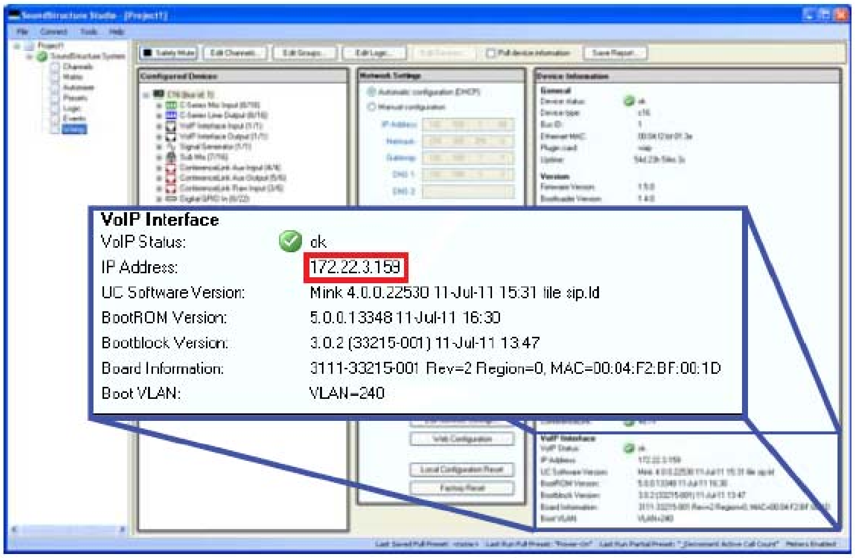
A microphone located on the ceiling.
Input Gain: 58 dB

If you don’t have the VoIP interface input and output virtual channels defined in the project, using the **Edit Channels** control select a VoIP Interface, click Add, and close the Edit Channels control. Use the **File > Save** option to save your file to ensure your new channel definition will survive a power cycle.

Valid IP Address

Check that you have a valid IP address for the SoundStructure VoIP Interface on the Wiring page as shown in the following figure.

SoundStructure VoIP IP Address in SoundStructure Studio



VoIP Interface

VoIP Status: ✓ ok

IP Address: 172.22.3.159

UC Software Version: Mink 4.0.0.22530 11-Jul-11 15:31 file sip.ld

BootROM Version: 5.0.0.13348 11-Jul-11 16:30

Bootblock Version: 3.0.2 (33215-001) 11-Jul-11 13:47

Board Information: 3111-33215-001 Rev=2 Region=0, MAC=00:04:F2:BF:00:1D

Boot VLAN: VLAN=240

If you don’t have a valid IP address, check whether the network cable for the SoundStructure VoIP Interface is plugged in and connected to the VoIP network.

If your cable is plugged in but you don’t have a valid IP address, check whether DHCP is enabled for the device and you have an DHCP server on your subnet.

If you don’t have a DHCP server on your local subnet, set a static IP address from the SoundStructure Studio wiring page with the Edit Network Connections button. See Setting an IP address with SoundStructure Studio.

Valid Network Route to the SoundStructure VoIP Interface

If you have a valid IP address, click the Web Configuration button to open the Web Configuration Utility.

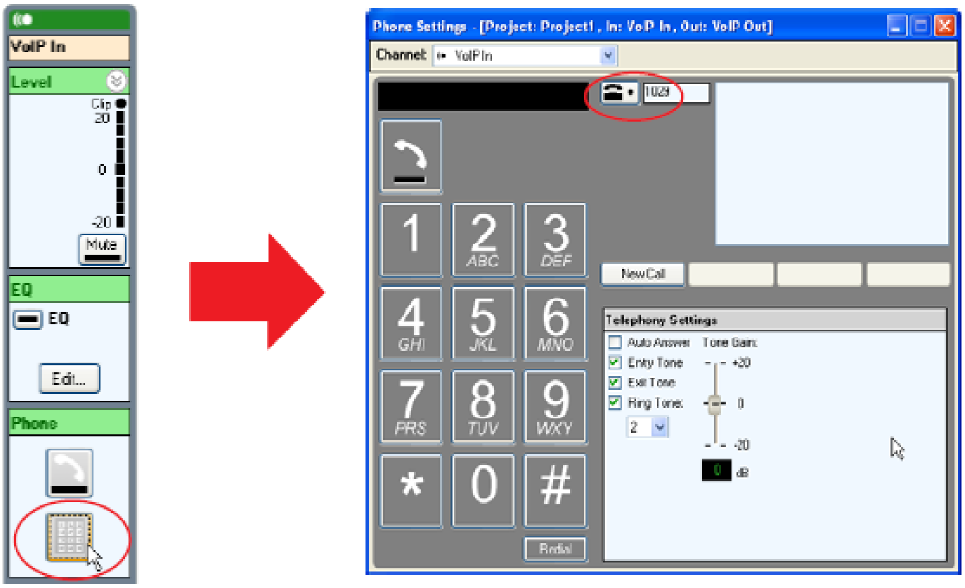
If you receive an IP address for the SoundStructure VoIP Interface but can’t get access to the Web Configuration Utility and can’t network ping the SoundStructure VoIP Interface, then it is possible the SoundStructure VoIP Interface is on a different VLAN or subnet than your computer and not all data traffic may be permitted between the two different subnets. Check with your local IT administrator if you need a route from your PC to the SoundStructure VoIP Interface for setup purposes.

If it is not possible to provide a route to the SoundStructure VoIP Interface’s IP address, you may need to temporarily insert a local router that can source DHCP addresses and connect both your PC and the SoundStructure VoIP Interface to the local router. From the PC you should be able to reach the SoundStructure VoIP Interface and configure the device via the Web Configuration Utility. Once done, you may reconnect the original network connections.

Confirming Line Registration

Navigate to the Channels page and open the Phone Settings dialog by clicking on the dialer control on the Channels page as shown on the left side of the following figure. On the phone settings page you should see a registered line icon if the line is properly registered (see right side of the figure). See Registering Lines with the SoundStructure VoIP Interface for information on how to register a line. In this example, extension 1029 has been defined and registered successfully.

Opening Phone Settings from Channels Page


VoIP In
Level
Clip
20
0
-20
Mute
EQ
EQ
Edit...
Phone
Phone Settings - [Project: Project1, In: VoIP In, Out: VoIP Out]
1 2 ABC 3 DEF
4 GHI 5 JKL 6 MNO
7 PRS 8 TUV 9 WXY
* 0 #
New Call
Redial
Telephony Settings
Tone Gain:
+20
0
-20
dB

Channel: VoIP In

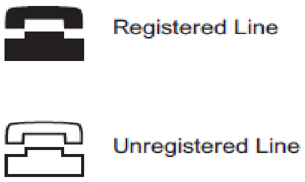
1029

☐ Auto Answer
☑ Entry Tone
☑ Exit Tone
☑ Ring Tone: 2

0 dB

The following figure shows the two states the line registration icon may have depending on whether the line is registered or unregistered. If your line is unregistered, confirm the registration settings via the Web Configuration Utility.

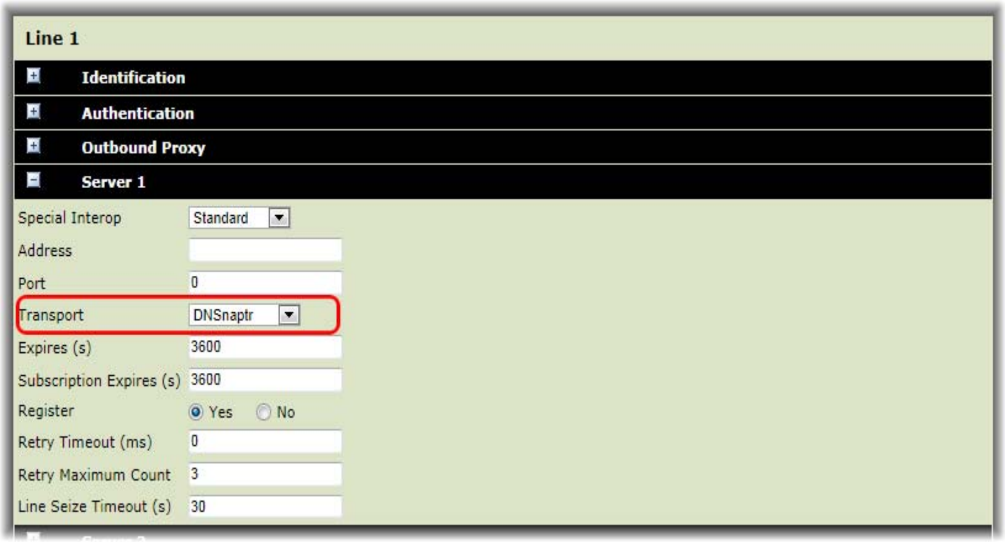
Line Registered and Unregistered Line Icons


Registered Line
Unregistered Line

In this example, the dark phone icon on the phone settings page indicates that extension 1029 has been properly registered.

If the line appears unregistered on the Phone Settings page, and on the Web Configuration Utility page the Identification settings are correct, check that the transport settings on the line settings page within the Web Configuration Utility is set correctly. Some networks may require TCPOnly.

Transport Settings



Line 1

⊞ Identification

⊞ Authentication

⊞ Outbound Proxy

⊟ Server 1

Special Interop: Standard

Address:

Port: 0

Transport: DNSnaptr

Expires (s): 3600

Subscription Expires (s): 3600

Register: ● Yes ○ No

Retry Timeout (ms): 0

Retry Maximum Count: 3

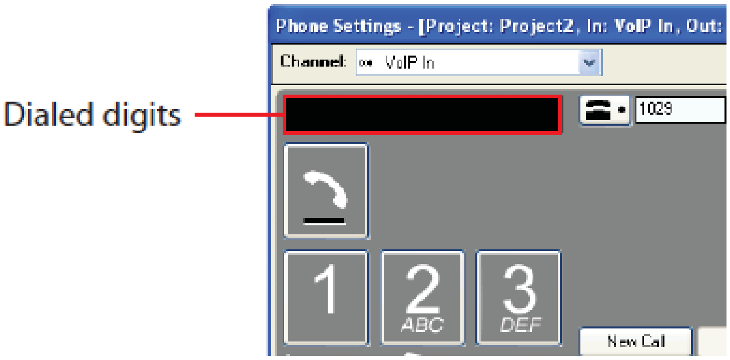
Line Seize Timeout (s): 30

Finally, check SoundStructure VoIP Interface Application log as described in VoIP Interface Logs for additional information as to why the registration is unsuccessful.

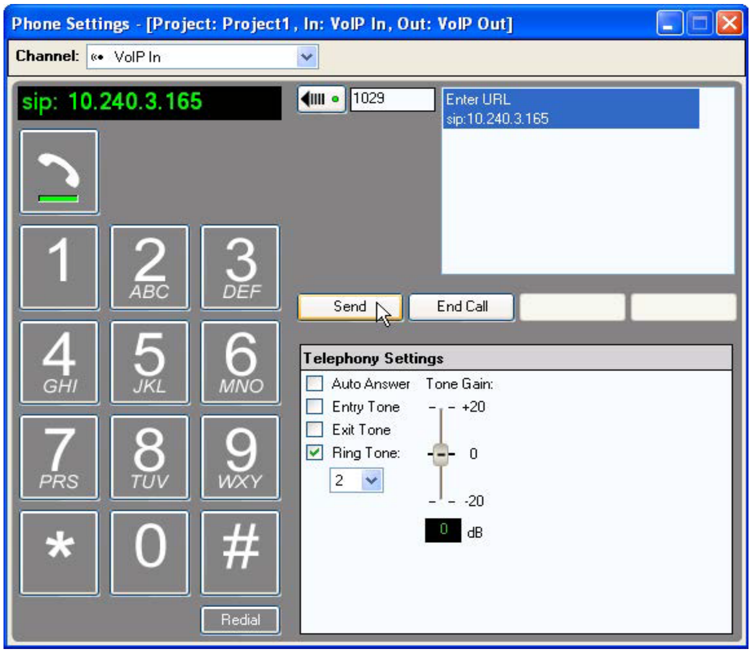
Dialing a SIP URL Call

If you are having trouble registering a line, it is possible to test that the SoundStructure VoIP Interface is installed properly, except for the line registration, by dialing a SIP URL call to another SIP device. To dial a SIP URL call:

1. From the Phone Settings Menu, select **New Call** and then press **URL**.
2. Click in the Dialed digits area as shown below and enter the URL to dial using your PC keyboard. For the 'dot' in between the IP address octets, use the '.' on your keyboard.



The remote endpoint can be a desktop phone or the IP address of a different SoundStructure VoIP Interface being installed. Once the IP address has been entered, click **Send** as shown next.

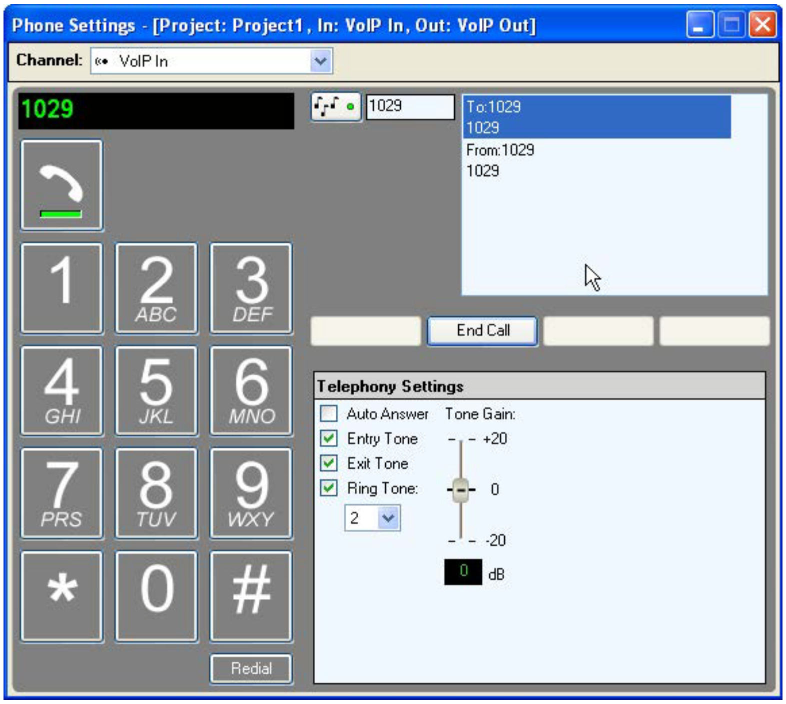

Phone Settings - [Project: Project1, In: VoIP In, Out: VoIP Out]
Channel: VoIP In
sip: 10.240.3.165
1029
Enter URL
sip:10.240.3.165
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PRS
8 TUV
9 WXY
*
0
#
Redial
Send
End Call
Telephony Settings
☐ Auto Answer
☐ Entry Tone
☐ Exit Tone
☑ Ring Tone:
2
Tone Gain:
+20
0
-20
0 dB

The remote SIP device will ring and a call may be connected if the remote device answers the telephone call. Once the call is connected, the SoundStructure VoIP Interface is installed properly except for line registration information.

Dialing a Call with a Registered Line

If you have a registered line, you should be able to press the **New Call** button to take the phone offhook. Next, dial the extension of another registered phone and press **Send**. If you have dialed a valid extension, you will hear the call ring. If you don't have a valid extension to dial, call your own extension and you will see an incoming call display as shown next where extension 1029 has dialed extension 1029

Call on a Registered Line



Phone Settings - [Project: Project1, In: VoIP In, Out: VoIP Out]

Channel: VoIP In

1029

1029

To:1029
1029
From:1029
1029

Telephony Settings

☐ Auto Answer
☑ Entry Tone
☑ Exit Tone
☑ Ring Tone: 2

Tone Gain: 0 dB

1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PRS
8 TUV
9 WXY
*
0
#
Redial
End Call
+20
0
-20

VoIP Interface Logs

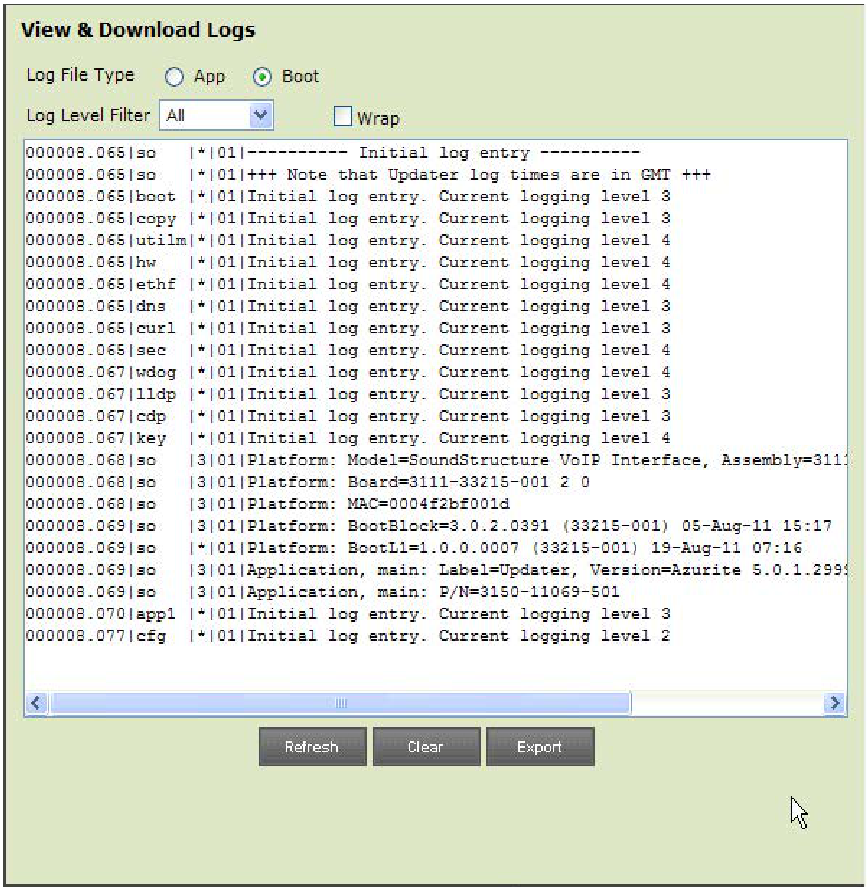
You may view the logs of the SoundStructure VoIP Interface with the Web Configuration Utility by navigating to the **Diagnostics > View & Download Logs.**

There are two types of logs available: Boot and App.

Boot Logs

Boot logs show information regarding the boot process of the SoundStructure VoIP Interface including any firmware updates. The boot log shown next depicts a system after it has booted up.

SoundStructure VoIP Interface Boot Log



```
000008.065|so   |*|01|---------- Initial log entry ----------
000008.065|so   |*|01|+++ Note that Updater log times are in GMT +++
000008.065|boot |*|01|Initial log entry. Current logging level 3
000008.065|copy |*|01|Initial log entry. Current logging level 3
000008.065|utilm|*|01|Initial log entry. Current logging level 4
000008.065|hw   |*|01|Initial log entry. Current logging level 4
000008.065|ethf |*|01|Initial log entry. Current logging level 4
000008.065|dns  |*|01|Initial log entry. Current logging level 3
000008.065|curl |*|01|Initial log entry. Current logging level 3
000008.065|sec  |*|01|Initial log entry. Current logging level 4
000008.067|wdog |*|01|Initial log entry. Current logging level 4
000008.067|lldp |*|01|Initial log entry. Current logging level 3
000008.067|cdp  |*|01|Initial log entry. Current logging level 3
000008.067|key  |*|01|Initial log entry. Current logging level 4
000008.068|so   |3|01|Platform: Model=SoundStructure VoIP Interface, Assembly=311
000008.068|so   |3|01|Platform: Board=3111-33215-001 2 0
000008.068|so   |3|01|Platform: MAC=0004f2bf001d
000008.069|so   |3|01|Platform: BootBlock=3.0.2.0391 (33215-001) 05-Aug-11 15:17
000008.069|so   |*|01|Platform: BootL1=1.0.0.0007 (33215-001) 19-Aug-11 07:16
000008.069|so   |3|01|Application, main: Label=Updater, Version=Azurite 5.0.1.299
000008.069|so   |3|01|Application, main: P/N=3150-11069-501
000008.070|app1 |*|01|Initial log entry. Current logging level 3
000008.077|cfg  |*|01|Initial log entry. Current logging level 2
```

App Logs

Application logs show information regarding the application running on the SoundStructure VoIP Interface. There are different levels of the log information that may be viewed by changing the value of the Log Level Filter.

The application logs can provide additional insight into why a line has not registered successfully. For example, if the authentication user name is not correct, the log will contain a message as follows:

```
1117164420|sip |4|00|Registration failed User: 1234, Error Code:404 Not Found
```

If a the call server is not available or the server IP address is not correct, the log may contain a message as follows:

```
1117161439|pmt |4|00|Login failure.
1117161441|sip |4|00|Registration failed User: 200024, Error Code:480 Temporarily not
available
```

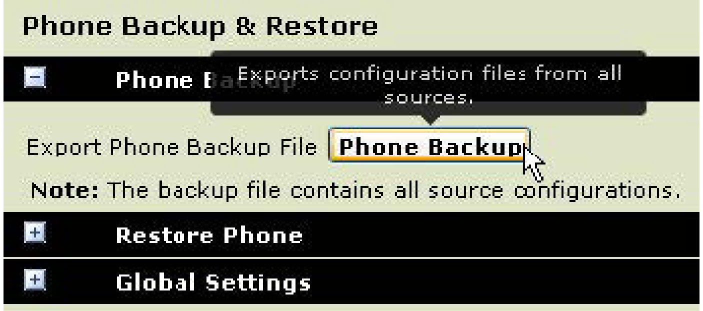
Back up and Restore the VoIP Specific Settings

The VoIP settings on the SoundStructure VoIP Interface may be backed up or restored by using Web Configuration Utility Phone Backup & Restore option.

Backup Settings

To backup the phone settings

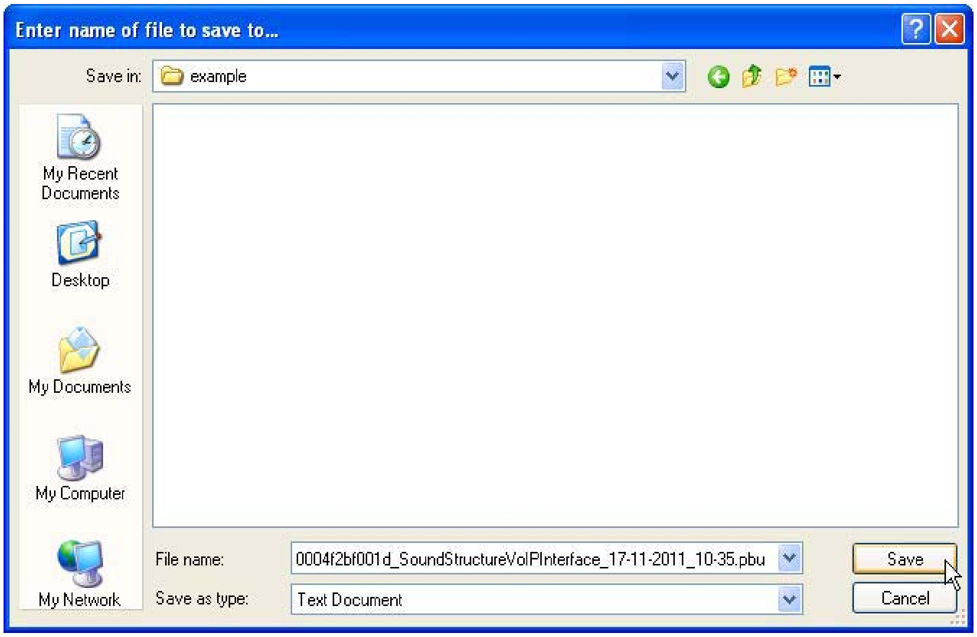
1 Select **Phone Backup** as shown in the following figure.


Phone Backup & Restore
Phone B
Export Phone Backup File
Phone Backup
Restore Phone
Global Settings

Exports configuration files from all sources.

Note: The backup file contains all source configurations.

2 Enter a filename to store the settings. The default filename includes the MAC address, the device type, and the current date and time as shown in the following figure.


Enter name of file to save to...
My Recent Documents
Desktop
My Documents
My Computer
My Network
Save
Cancel

Save in: example

File name: 0004f2bf001d_SoundStructureVoIPInterface_17-11-2011_10-35.pbu

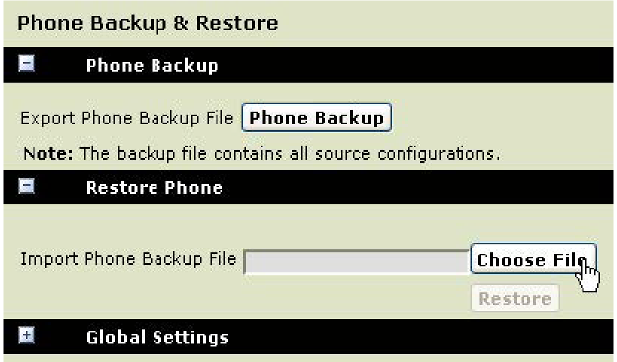
Save as type: Text Document

3 Click **Save** to store the settings in a text file that can be viewed with any text editor.

Restore Settings

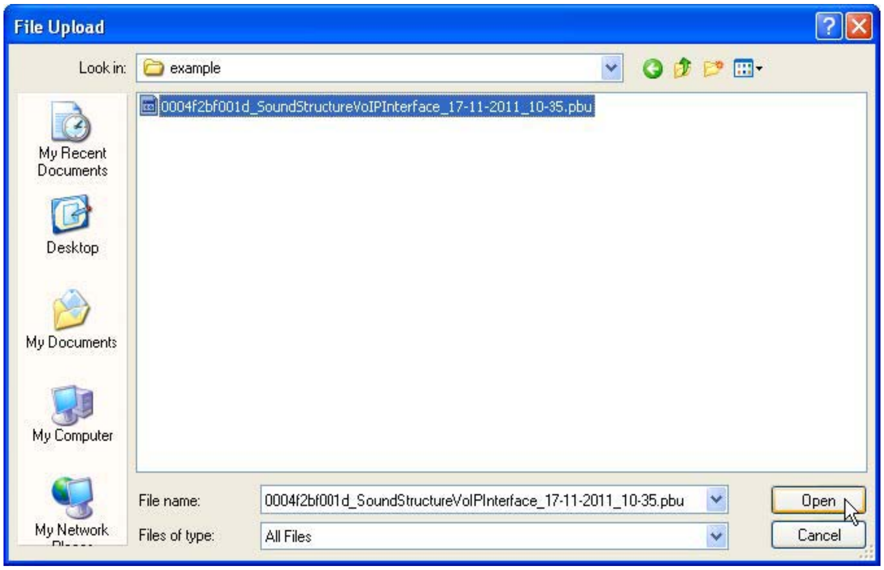
The settings for the SoundStructure VoIP Interface can be restored by clicking the **Choose File** button as shown in the following figure.

Restoring Phone Settings


Phone Backup & Restore
Phone Backup
Export Phone Backup File
Phone Backup
Restore Phone
Import Phone Backup File
Choose File
Restore
Global Settings

Note: The backup file contains all source configurations.

Clicking **Choose File** will prompt you to select the filename to restore as shown in the following figure.


File Upload
Look in:
example
0004f2bf001d_SoundStructureVoIPInterface_17-11-2011_10-35.pbu
My Recent Documents
Desktop
My Documents
My Computer
My Network Places
Open
Cancel

File name: 0004f2bf001d_SoundStructureVoIPInterface_17-11-2011_10-35.pbu

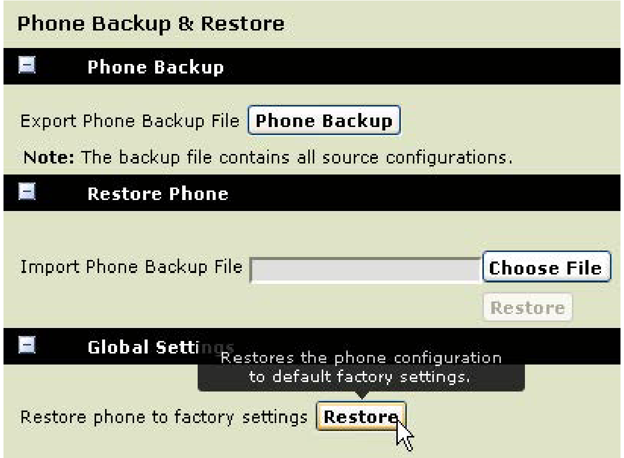
Files of type: All Files

Once the file is opened, the settings will be restored to the SoundStructure VoIP Interface.

Global Settings

The SoundStructure VoIP Interface settings may be reset to factory default values by clicking the **Restore** button in the Global Settings section as shown in the following figure.

Restoring Global Settings


Phone Backup & Restore
Phone Backup
Export Phone Backup File
Phone Backup
Restore Phone
Import Phone Backup File
Choose File
Restore
Global Settings
Restore phone to factory settings
Restore

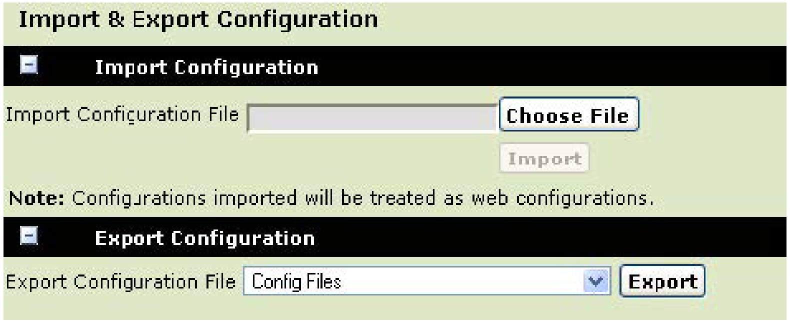
Note: The backup file contains all source configurations.

Restores the phone configuration to default factory settings.

Importing and Exporting VoIP Parameter Settings

You can use the Web Configuration Utility to export the VoIP specific configuration settings of the SoundStructure VoIP Interface. To import settings, navigate to the **Utilities > Import & Export Configuration** and click **Choose File** and then **Import** as shown next.

Importing Configuration Files


Import & Export Configuration
Import Configuration
Import Configuration File
Choose File
Import
Export Configuration
Export Configuration File
Config Files
Export

Note: Configurations imported will be treated as web configurations.

You can export different settings from the SoundStructure VoIP Interface as shown next.

Exported SoundStructure VoIP Interface Settings

Export Configuration File	Description
All Configuration Settings (except Device Settings)	Exports the configuration from all sources except for the device specific settings including line registration information, codec preferences, and other information that has been configured differently from the default values.
Config Files	Exports the parameters from the configuration files which includes line registration information. This file is useful if you want to take the line registration from one system, customize it, and reuse it in a different system.

Exported SoundStructure VoIP Interface Settings

Export Configuration File	Description
Local	Exports parameters that were configured on the phone’s local user interface. This is not relevant to the SoundStructure VoIP Interface.
Web	Exports parameters setup in the Web Configuration Utility including codec priorities and other changes from the default configuration exclusive of line registration information.
Device Settings	Exports the parameters associated with the device including DHCP options, provisioning server authentication settings, and more.

To export a configuration, choose the type of export and click **Export**. If you are interested in exporting the line registration information, you will want to select the Config File option.

SoundStructure Log Information

You can get the SoundStructure logs from the Wiring page within SoundStructure Studio.

If the SoundStructure VoIP Interface is detected in a system, the boot section of the logs will show that a card was detected and in which SoundStructure system the card was found. An example of this is shown next.

```
Sept 2 05:25:11 dsploader: sts: Detected plug-in card type: VOIP
...
Sept 2 05:25:36 gcp: sts: discovered 1 devices
Sept 2 05:25:36 gcp: sts: 1: c16 [voip ]
Jul 28 05:25:36 gcp: sts: starting global command processor
Jul 28 05:25:37 lcp: sts: waiting for VoIP plug-in card to boot
Jul 28 05:25:38 gcp: sts: parsed 273 parameter definitions from
/usr/share/serendipity/params.xml
Jul 28 05:25:38 gcp: sts: parsed 24 meter definitions from
/usr/share/serendipity/meters.xml
Jul 28 05:25:38 gcp: sts: connected 24 meter handlers
Jul 28 05:25:38 gcp: sts: connected 118 RAF-mapped parameters from
/usr/share/serendipity/rafmap.xml
Jul 28 05:25:38 gcp: sts: connected 0 dummy parameters
Jul 28 05:25:40 lcp: sts: VoIP plug-in card booted successfully
```

At the end of the SoundStructure system log there will be a summary (shown next) of the SoundStructure information and the SoundStructure VoIP Interface settings.

```
========================================
============== system info =============
========================================
sys_name: SoundStructure System
sys_num_devs: 1
                  dev_bus_id                 1
```

```
                    dev_type               c16
             dev_plugin_type              voip
                  dev_status                ok
                  dev_uptime        4:03:58:38
                     eth_mac 00:04:f2:bf:01:3e
         eth_settings [mode]              dhcp
         eth_settings [addr]      192.168.1.85
          eth_settings [dns]       192.168.1.1
           eth_settings [gw]       192.168.1.1
           eth_settings [nm]     255.255.255.0
                    ser_baud              9600
                    ser_flow              none
            dev_firmware_ver             1.5.0
          dev_bootloader_ver             1.4.0
            dev_habanero_ver          B4121415
                  dev_hw_rev                 A
                  dev_hw_eco                 2
  clink_num_attached [table]                 0
clink_num_attached [ceiling]                 1
 clink_num_attached [codecs]                 1
                dev_temp [1]              41.3
                dev_temp [2]              65.8
                dev_temp [3]              34.9
        dev_volt_phantom [1]              47.3
        dev_volt_phantom [2]              47.3
        dev_volt_phantom [3]              47.7
        dev_volt_phantom [4]              48.1
             dev_volt_pos_15              14.7
             dev_volt_neg_15             -14.9
              dev_volt_clink              48.1

VoIP Plug-In Card 1
             voip_board_info 3111-33215-001 Rev=2 Region=0, MAC=00:04:F2:BF:00:1D
       voip_bootblock_sw_ver 3.0.2.0391 (33215-001) 05-Aug-11 15:17
         voip_bootrom_sw_ver 5.0.1.2999 16-Sep-11 07:05
              voip_uc_sw_ver Mink 4.0.1.4394 16-Sep-11 06:41 file sip.ld
         voip_dhcp_boot_serv            static
     voip_dhcp_boot_serv_opt               160
    voip_dhcp_boot_serv_type            string
    voip_dhcp_option_60_type      ascii_string
    voip_eth_settings [mode]              dhcp
    voip_eth_settings [addr]     192.168.1.118
     voip_eth_settings [dns] 192.168.1.1 0.0.0.0
      voip_eth_settings [gw]       192.168.1.1
      voip_eth_settings [nm]     255.255.255.0
            voip_eth_vlan_id                -1
      voip_prov_serv_address     192.168.1.200
         voip_prov_serv_type               ftp
         voip_prov_serv_user          username
                 voip_status                ok
```

Information Required for Support

Polycom’s Technical Support team is ready to help our integration partners ensure their equipment is functioning properly. For specific questions regarding integration with a VoIP PBX or call management platform, you should contact the IT/Phone integration specialist in your organization or at your end user’s installation.

Before calling Polycom Technical Support for assistance, Polycom recommends that you have the following information:

- SoundStructure project file (.str extension).

  This file allows the support team to understand the design and see if the SoundStructure VoIP Interface is part of the project.
- SoundStructure device log (retrieved from the SoundStructure Studio Wiring page)

  This file allows the support team understand the versions of firmware and other settings of the SoundStructure device.
- Whether you have been able to make a SIP URL call to another SIP device.

  This helps separate baseline functionality questions from PBX integration questions. For PBX specific integration questions about line registration problems to your local PBX, consult your local IT/Phone specialist who has deployed other VoIP endpoints in that environment.

Understanding SoundStructure VoIP Interface API Commands

The existing SoundStructure API commands have been updated to support the SoundStructure VoIP Interface input and output channels. There are also new API commands that have been created for the SoundStructure VoIP Interface. For a quick overview of the commands, refer to the table shown next. For more complete information, see the updated command set in Appendix A: Command Protocol Reference Guide in Appendix A or navigate directly to the SoundStructure device with your PC’s browser.

SoundStructure VoIP Interface API Commands

Parameter Name	Description
voip_answer	Used to answer an incoming call.
voip_blind	Used with voip_transfer to make a blind transfer.
voip_board_info	Returns manufacturing and hardware information about the plug-in card.
voip_bootblock_sw_ver	Returns the bootblock version of the plug-in card.
voip_bootrom_sw_ver	Returns the bootROM version of the plug-in card.
voip_call_appearance	Selects the currently active call appearance. Some parameters operate on the currently active call appearance. This is similar to pressing a call appearance on a desktop phone.

SoundStructure VoIP Interface API Commands

voip_call_appearance_info	Reports textual caller-id information for the specified call appearance.
voip_call_appearance_line	Reports the line number associated with the specified call appearance.
voip_call_appearance_state	Reports the state of the specified call appearance. Examples states include ‘connected’, ‘ringback’, ‘disconnected’, and others.
voip_cancel	Used to cancel a transfer or a conference.
voip_conference	Used to start a conference call.
voip_dhcp_boot_serv	Controls the boot server option for the plug-in card.
voip_dhcp_boot_serv_opt	When voip_dhcp_boot_serv is set to custom, this parameter specifies the DHCP option number in which the plug-in card will look for the boot server.
voip_dhcp_boot_serv_type	When voip_dhcp_boot_serv is set to custom, this parameter specifies the type of the DHCP option in which the plug-in card will look for the boot server.
voip_dhcp_option_60_type	Specifies the format for the vendor identifying information used with a DHCP server when DHCP option 60 is enabled.
voip_dial_mode	Selects between number dialing and SIP URL dialing.
voip_dnd	Enables or disables the Do Not Disturb mode on all lines.
voip_eth_settings	Sets or gets the Ethernet settings for the plug-in card.
voip_eth_vlan_id	Sets or gets the VLAN ID for the plug-in card. A value of -1 corresponds to no vlan id.
voip_factory_reset	Resets the plug-in card to its factory state.
voip_forward	Used to forward a call.
voip_hold	Places the current call on hold.
voip_join	Creates a conference from two calls.
voip_line	Selects a line. This is similar to pressing a line key on a desktop phone.
voip_line_label	Reports the label for the specified line.
voip_line_state	Reports the state for the specified line, e.g., registered, not registered, active, conference, hold, etc.
voip_loc_city[1]	Displays the city field when the Lync Server is configured with location information. Corresponds with the `loc.Info.x.A3` Polycom UC software parameter.
voip_loc_company_name[1]	Displays the company name field when the Lync Server is configured with location information. Corresponds with the `loc.Info.x.NAM` Polycom UC software parameter.

SoundStructure VoIP Interface API Commands

voip_loc_country[1]	Displays the country field when the Lync Server is configured with location information. Corresponds with the `loc.Info.x.country` Polycom UC software parameter.
voip_loc_description[1]	Displays the description field when the Lync Server is configured with location information. Corresponds with the `loc.Info.x.label` Polycom UC software parameter.
voip_loc_house_number[1]	Displays the house number field when Lync Server is configured with location information. Corresponds with the `locInfo.x.HNO` Polycom UC software parameter.
voip_loc_location[1]	Displays the additional location field when Lync Server is configured with location information. Corresponds with the `locInfo.x.LOC` Polycom UC software parameter.
voip_loc_postal_code[1]	Displays the postal code field when Lync Server is configured with location information. Corresponds with the `locInfo.x.PC` Polycom UC software parameter.
voip_loc_pre_directional[1]	Displays the pre directional field when Lync Server is configured with location information. Corresponds with the `locInfo.x.PRD` Polycom UC software parameter.
voip_loc_state[1]	Displays the state field when Lync Server is configured with location information. Corresponds with the `locInfo.x.A1` Polycom UC software parameter.
voip_loc_street_name[1]	Displays the street name field when Lync Server is configured with location information. Corresponds with the `locInfo.x.RD` Polycom UC software parameter.
voip_loc_street_suffix[1]	Displays the street suffix field when Lync Server is configured with location information. Corresponds with the `locInfo.x.STS` Polycom UC software parameter.
voip_local_reset	Resets all the VoIP parameters that SoundStructure Studio can set: ring type, auto answer, volume of the auto answer tone.
voip_message_waiting	Indicates whether voice mail messages from any registered lines are waiting for the SoundStructure VoIP Interface
voip_net_cfg_save	Causes the VoIP network settings to be written to the flash memory.
voip_notification[1]	Displays automatic status updates retrieved from the SoundStructure VoIP Interface.
voip_popup[1]	Displays immediate status notifications retrieved from the SoundStructure VoIP Interface.
voip_prov_serv_address	Sets the address of the provisioning server for the VoIP interface.
voip_prov_serv_password	Sets the password for the provisioning server for the VoIP interface.
voip_prov_serv_type	Sets the type of provisioning server for the plug-in card.

SoundStructure VoIP Interface API Commands

voip_prov_serv_user	Sets the username for logging into the provisioning server.
voip_reboot	Reboots the SoundStructure VoIP Interface.
voip_resume	Resumes a call that was on hold.
voip_send	Causes a call to be placed with the digits dialed.
voip_split	Splits all calls in a conference into individual calls.
voip_status	Indicates the status LED state on the SoundStructure VoIP Interface.
voip_transfer	Used to transfer a call.
voip_uc_sw_ver	Gets the SoundStructure VoIP Interface’s UC Software version.

1. These SoundStructure API commands were added in SoundStructure Firmware 1.7.0 and UC software 4.1.13G.

Using the SoundStructure API

This section presents some examples of using the SoundStructure API commands. In this section, commands sent by the control system are in blue, acknowledgments received by the control system are preceded by the word ‘val’ and are coded in green, and comments to explain the example are prefaced by the ‘#’ character.

Not all acknowledgments are shown in these examples. Only the acknowledgments that are important for understanding how the systems works are shown. These examples assume the names of the SoundStructure VoIP Interface’s input and output channel are “VoIP In” and “VoIP Out” respectively.

Dialing a Call

This example shows how to make a phone call.

# Take the phone offhook.

```
set phone_connect “VoIP Out” 1
val phone_connect “VoIP Out” 1
```

# Dial the digits

```
set phone_dial “VoIP Out” “2029”
val phone_dial “VoIP Out” “2029”
```

# If the remote party answers at extension 2029 then the call

# is connected.

# Alternatively, you can now dial the digits first and as

# long as you take the phone off hook within 20 seconds of dialing,

```
# the digits will be retained and dialed.

set phone_dial "VoIP Out" "2029"
val phone_dial “VoIP Out” “2029”
# Take the phone offhook which will cause the digits to be dialed
# if the phone_connect command is sent within 20 seconds of the
# phone_dial command
set phone_connect "VoIP Out" 1
val phone_connect “VoIP Out” 1
```

Hanging up a Call

This example shows how to hang up a phone call.

```
# To hang up the phone
set phone_connect “VoIP Out” 0
val phone_connect “VoIP Out” 0
```

Putting a Call on Hold and Resuming the Call

This example shows how to dial a call, place the call on hold, and resume the existing call.

```
# Take the phone offhook
set phone_connect “VoIP Out” 1
val phone_connect “VoIP Out” 1
# Dial the digits of the initial call
set phone_dial “VoIP Out” “2029”
val phone_dial “VoIP Out” “2029”
# Once you know the call is connected by waiting for the
# voip_call_appearance_state set to connected
val voip_call_appearance_state "VoIP Out" 1 connected
# Then you can place the connected call on hold
set voip_hold “VoIP Out”
# Once the call is on hold, you will get the confirmation with the
# voip_call_appearance_state acknowledgment
val voip_call_appearance_state "VoIP Out" 1 ncas_call_hold
# To resume the call, use the voip_resume command
set voip_resume “VoIP Out”
# Once you have the acknowledgment that the line is connected,
```

# then the call has resumed.

```
val voip_call_appearance_state "VoIP Out" 1 connected
```

# To hang up the phone

```
set phone_connect “VoIP Out” 0
```

Forwarding an Incoming Call

This example shows how to take an incoming call and forward it to different extension. In this example, an incoming call from extension 2029 rings in the local room and is forwarded to extension 5148.

# Incoming phone call generates a series phone_ring messages on

# the “VoIP In” channel

```
val phone_ring "VoIP In" 1
```

# To initiate the forward, send the voip_forward command

```
set voip_forward “VoIP Out”
val voip_forward "VoIP Out"
```

# dial the extension to transfer the call to

```
set phone_dial "VoIP Out" "5148"
val phone_dial "VoIP Out" "5148"
```

# Complete the forward by sending voip_forward again

```
set voip_forward "VoIP Out"
val voip_forward "VoIP Out"
```

# the call appearance information is cleared as the call is forwarded

# to the next extension

```
val voip_call_appearance_info "VoIP Out" 1 1 "2029"
val voip_call_appearance_info "VoIP Out" 1 1 ""
val voip_call_appearance_info "VoIP Out" 1 2 ""
val voip_call_appearance_state "VoIP Out" 1 free
val voip_line_state "VoIP Out" 1 reg
```

# the local phone stops ringing as the call is forwarded.

```
val phone_ring "VoIP In" 0
```

Transferring a Call

This example shows how you can dial a call, connect to the remote party (extension 2029 in this example), and then perform a consultative transfer of that call to a different extension (extension 5148 in this example).

# Dial the desired number

```
set phone_dial "VoIP Out" "2029"
val phone_dial “VoIP Out” “2029”
# Take the phone offhook to send the digits to the call server
set phone_connect "VoIP Out" 1
val phone_connect “VoIP Out” 1
# Once connected the call appearance state will be updated to connected
val voip_call_appearance_state “VoIP Out” 1 connected
# Once you are connected, then initiate the transfer.
# This will generate new dial tone.
set voip_transfer "VoIP Out"
# Once you received an indication that you have dialtone you can
# dial the digits
val voip_call_appearance_state "VoIP Out" 1 dialtone
# Dial the party to whom you would like to transfer the call
set phone_dial "VoIP Out" "5148"
# Optionally use the voip_send command to send the digits immediately
# without waiting for a time-out from the dial plan
set voip_send "VoIP Out"
# Once you are connected, you can then talk to the person to tell
# them you are transferring the call
val voip_call_appearance_state "VoIP Out" 1 connected
# Then once connected, you complete the transfer
set voip_transfer "VoIP Out"
```

Blind Transfer of a Call

This example shows how you can dial a call and connect to the remote party (extension 2029 in this example) and then ‘blind transfer’ that call to a different extension (5148 in this example). When you ‘blind transfer’ the call, you do not have to establish the call to the receiving party before transferring the call.

```
# Dial the desired number
set phone_dial "VoIP Out" "2029"
val phone_dial “VoIP Out” “2029”
# Take the phone offhook to send the digits to the call server
set phone_connect "VoIP Out" 1
val phone_connect “VoIP Out” 1
```

```
# Once connected the call appearance state will be updated to connected
val voip_call_appearance_state “VoIP Out” 1 connected
# Once you are connected, then initiate the transfer.
# This will generate new dial tone.
set voip_transfer "VoIP Out"
# set the transfer to be a blind transfer
set voip_blind “VoIP Out”
# Once you received an indication that you have dialtone you can
# dial the digits
val voip_call_appearance_state "VoIP Out" 1 dialtone
# Dial the party to transfer the call
set phone_dial "VoIP Out" "5148"
# Optionally use the voip_send command to send the digits immediately
# without waiting for a dial time-out from the dial plan
set voip_send "VoIP Out"
# Once the blind transfer is enabled, the initial party is
# connected to the ringing call and the SoundStructure VoIP Interface
# hangs up the local call
val voip_call_appearance_state "VoIP Out" 1 ncas_call_transfer
# local call is hung up
val phone_connect “VoIP Out” 0
```

Dialing Two Calls on the Same Line

This example shows how you can dial and bridge together two calls on the same line.

```
# Take the phone offhook
set phone_connect “VoIP Out” 1
# Dial the digits of the first call
set phone_dial “VoIP Out” “2029”
# Once you know the call is connected by waiting for the
# voip_call_appearance_state set to connected
val voip_call_appearance_state "VoIP Out" 1 connected
# Then you can use the conference feature to place the connected
# call on hold and get a new dialtone
set voip_conference “VoIP Out”
```

```
# Dial the second call and tell the system to dial with the
# voip_send command
set phone_dial “VoIP Out” “5148”
# a voip_send command will tell the call management to use these digits
# without waiting for any dialplan time-out
set voip_send “VoIP Out”
# Once the call is connected with the call_appearance state
val voip_call_appearance_state "VoIP Out" 1 connected
# Then send voip_conference again to merge all the parties together
set voip_conference “VoIP Out”
# Once you have the acknowledgment that the line has been answered
# by the remote party then the call has been conferenced.
val voip_call_appearance_state "VoIP Out" 1 ncas_call_conference
# To hang up the conference call phone. If either party hangs up
# first, you will still be connected to the other remote party until
# you hang up the phone
set phone_connect “VoIP Out” 0
val phone_connect “VoIP Out” 0
```

Dialing Two Calls on Different Lines

This example shows how to use two independent lines and to conference together the two lines to form a three-way conference call. This example assumes I have Line 1 and Line 2 registered with call management servers.

```
# Dial the digits of the first call - by default line 1 is dialed
set phone_dial “VoIP Out” “2029”
# Take the phone offhook to force the digits to be dialed
set phone_connect “VoIP Out” 1
# You know the call is connected when you receive the message
# voip_call_appearance_state is set to connected
val voip_call_appearance_state "VoIP Out" 1 connected
# Then you can put the first call on hold
set voip_hold “VoIP Out”
# Once the call is on hold, the call appearance state will change
# to ncas_call_hold
```

```
val voip_call_appearance_state "VoIP Out" 1 ncas_call_hold
# Dial the second call
set phone_dial “VoIP Out” “1117”
# Now select the second line which will cause the digits to be dialed
set voip_line “VoIP Out” 2
# Wait for the call to be connected
val voip_call_appearance_state "VoIP Out" 2 connected
# And then join the two call appearances together
set voip_join “VoIP Out
# once the conference call is completed, the call appearance state will
# change
val voip_call_appearance_state “VoIP Out” 2 ncas_call_conference
# To hang up the conference call phone. If either party hangs up
# first, you will still be connected to the other remote party until
# you hang up the phone
set phone_connect “VoIP Out” 0
val phone_connect “VoIP Out” 0
```

SoundStructure API Behavior Changes

This section reviews changes made to the SoundStructure API as a result of supporting the SoundStructure VoIP Interface. The behavior of the following commands have changed:

phone_dial

In version 1.5.0 of the SoundStructure firmware, the behavior of the phone_dial command has changed to store the dialed digits when the phone is onhook and to dial those digits once the phone is taken offhook if the phone is taken offhook within 20 seconds of dialing the digits. If you send subsequent phone_dial commands while the phone is onhook, the digits will be appended to the previous set dialed digits. After 20 seconds with no phone_dial or phone_connect commands, the dialed digit buffer will be cleared.

In previous versions of SoundStructure firmware a phone_dial command issued when the phone was onhook would be ignored and digits would neither be dialed nor stored for subsequent dialing.

There is no change to the phone_dial behavior if the phone was already offhook when the phone_dial command is issued.

To dial a SIP URL call, the digits must be dialed while phone_connect is set to 0. The phone_connect command can then be sent to take the phone offhook and cause the digits to be dialed. Dialing a SIP URL when offhook is not supported.

run Preset

The run action for presets has been enhanced to provide an immediate “run Preset” acknowledgement once the preset begins execution. After the preset has finished executing, the “ran Preset” acknowledgment is generated. This enhancement allows both SoundStructure Studio and control system applications to know that a preset is being executed and control system programmers can make a preset button inactive after the initial “run” acknowledgment is received and then make the preset button active again once the final “ran” acknowledgment has been received to prevent users from pressing a preset button multiple times for presets that take longer to run than expected.

# Execute a preset with the “run” syntax.

```
run “My Preset”
```

# Immediately SoundStructure will send an acknowledgment that

# the preset has begun executing

```
run “My Preset”
```

# Once the preset has finished executing, SoundStructure continues

# to send the final “ran” acknowledgment

```
ran “My Preset”
```

sys_factory_reset

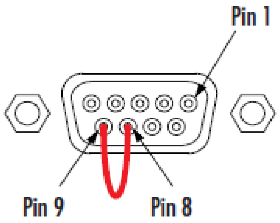
The command sys_factory_reset has been updated to also reset to factory defaults any SoundStructure VoIP Interfaces that may be installed in the SoundStructure system.

```
set sys_factory_reset
```

```
val sys_factory_reset
```

A system may also be reset to factory defaults by connecting pins 8 and 9 on the RS232 interface and applying power to the system. Once the system has finished booting, the connection from pins 8 to 9 can be removed. The RS232 port with pins 8 and 9 shorted together is shown next.

Pins 8 and 9 on the RS 232 Interface



Adding Authentication to SoundStructure Systems

In this chapter you will learn how to add authentication to SoundStructure system to prevent unauthorized access to SoundStructure systems. The SoundStructure authentication system allows you to add an administrator password to a SoundStructure system, change the password, and enable or disable authentication for a SoundStructure system.

SoundStructure Authentication Overview

The SoundStructure authentication feature allow two operating modes of the SoundStructure system: open and authenticated. The factory default setting is open.

Open systems do not require authentication before sending/receiving a configuration file or before controlling the system using the SoundStructure API. On open systems, SoundStructure API commands are sent over TCP on port 52774 as with all previous versions of SoundStructure software. By default SoundStructure systems operate in an open mode.

Authenticated systems require password authentication before being able to control the SoundStructure system over Ethernet, transfer configuration files, or update firmware. The password requirement prevents unauthorized access to, and control of, a SoundStructure system on a network connection. On authenticated systems, SoundStructure API commands are sent over TCP on port 52775 and once a system is authenticated, TCP communication over port 52774 is disabled. There is no authentication support on the RS-232 interface as the RS-232 interface always operates in an open mode.

When an authentication password is entered to the system, it is transmitted in clear text to the SoundStructure system over the network connection. The authentication password is stored locally on the SoundStructure system and is independent of the SoundStructure configuration file. Separating the password from the configuration file allows you to create and share SoundStructure configuration files without sharing the SoundStructure system password.

SoundStructure System Requirements

To use SoundStructure authentication, the following versions of software are required. To get the latest software versions, visit SoundStructure Support to download the required versions of software for your SoundStructure system.

SoundStructure Firmware version 1.6

This firmware version is fully compatible with configuration files created with earlier versions of SoundStructure Studio.

SoundStructure Studio version 1.8

This studio version is fully compatible with configuration files created with earlier versions of SoundStructure Studio.

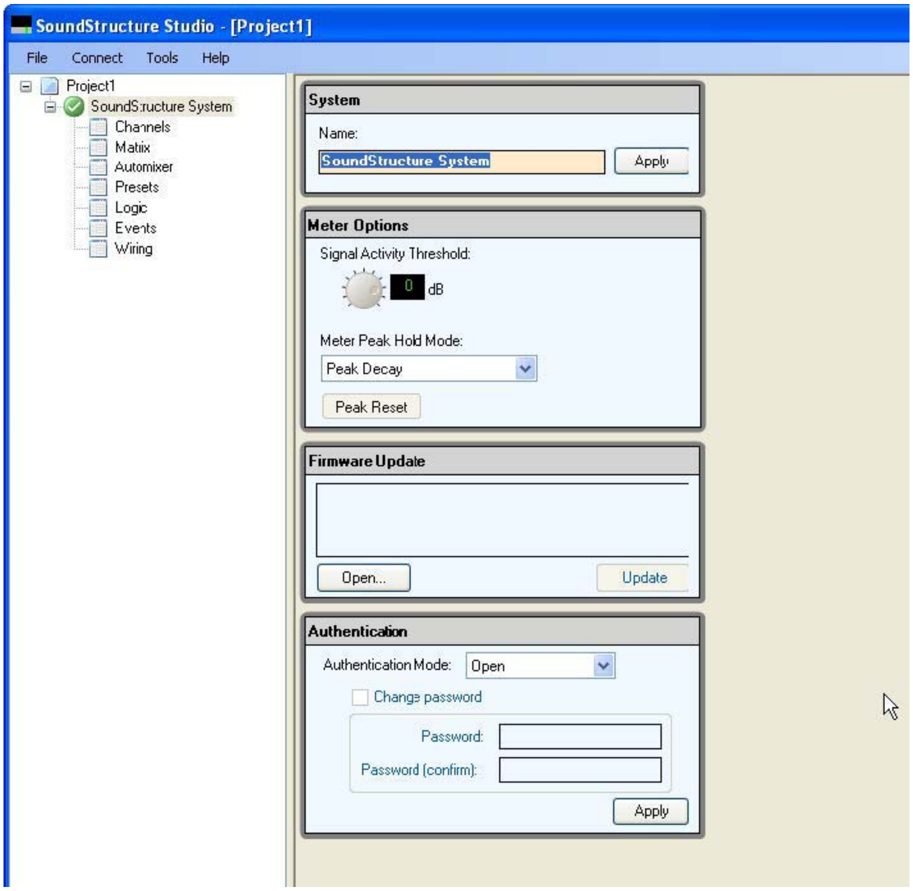
Configuration files created with SoundStructure Studio 1.8 are not compatible with older versions of SoundStructure Studio because there are new parameters defined to support authentication.

Enabling Authentication on a SoundStructure System

To use the authentication features you must update the SoundStructure system firmware to version 1.6 or newer and connect to the system using SoundStructure Studio version 1.8 or newer. Instructions for updating firmware are provided in Installing SoundStructure Devices.

Once connected to the SoundStructure System, follow these steps.

1 Navigate to the system page by clicking on the system name, **SoundStructure System**, as shown in the following figure.


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring
System
Apply
Meter Options
Peak Reset
Firmware Update
Open...
Update
Authentication
Change password
Password:
Password (confirm):
Apply

Name: SoundStructure System

Signal Activity Threshold: 0 dB

Meter Peak Hold Mode: Peak Decay

Authentication Mode: Open

☐ Change password

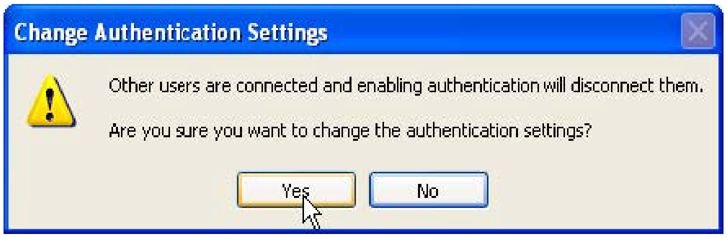
2 Change the Authentication mode from Open to **Authenticated**.

3 Optionally enable the **Change Password** control to enter a new password. The default password for the system is **456.**

   To change the default password, enter a new password and re-enter the password to confirm the password. Any string of characters and digits may be used. There are no minimum character password requirements. The password must no longer than 128 characters.

4 Click **Apply** to enable authentication.

5 If there are any other network connections over port 52774 to the SoundStructure system, you will be prompted whether you want to terminate those network connections and continue enabling authentication as shown in the following figure.



Change Authentication Settings

Other users are connected and enabling authentication will disconnect them.

Are you sure you want to change the authentication settings?

   To enable authentication, click **Yes**. The SoundStructure device will terminate any other API sessions connected over port 52774, including any control system connections over network port 52774. SoundStructure Studio will reconnect to the SoundStructure system over port 52775.

   RS232 API connections will not be terminated as the serial port always operates in open mode.

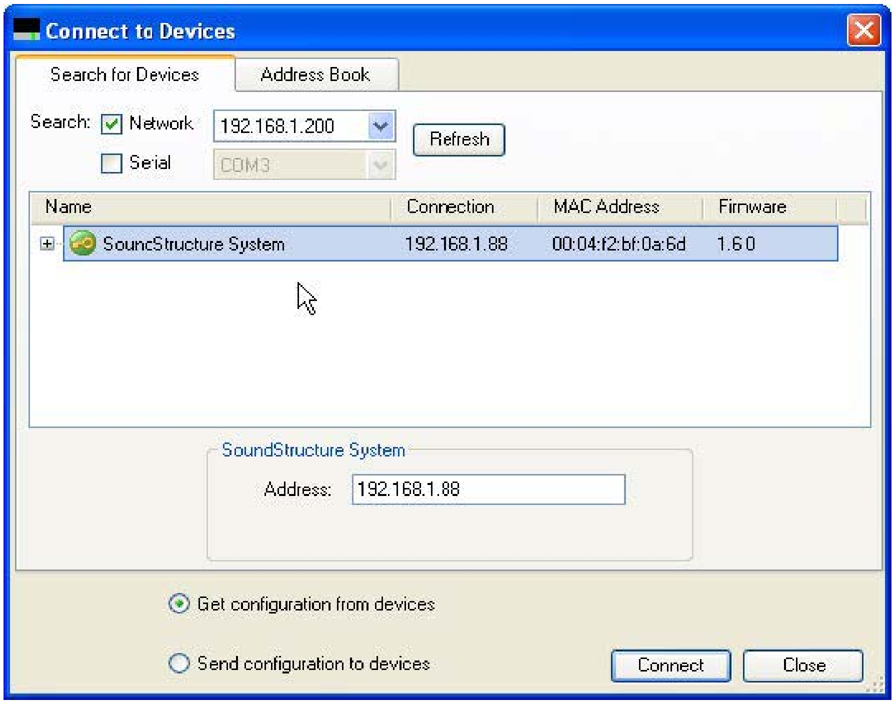
Discovering a System with Authentication

Once authentication has been enabled on a SoundStructure System, network communications over port 52774 are not allowed. Network communication must be performed over port 52775 for an authenticated system.

A SoundStructure system with authentication enabled is not discoverable by versions of SoundStructure Studio earlier than version 1.8 because older versions of SoundStructure Studio are only able to connect to the system over port 52774. SoundStructure Studio version 1.8 or newer can discover and connect to an authenticated system.

Using SoundStructure Studio version 1.8 or newer, a system may be discovered by selecting **Connect > Search for Devices.** Assuming your computer running SoundStructure Studio is on the same subnet as the desired SoundStructure systems, the discovery process will find SoundStructure devices as shown in the following figure. If you are not on the same subnet you can manually connect to the system by entering the IP address or selecting the system from the address book.

SoundStructure Studio Discovery Process



Search: ☑ Network 192.168.1.200

☐ Serial COM3

Name	Connection	MAC Address	Firmware
SouncStructure System	192.168.1.88	00:04:f2:bf:0a:6d	1.6.0

SoundStructure System

Address: 192.168.1.88

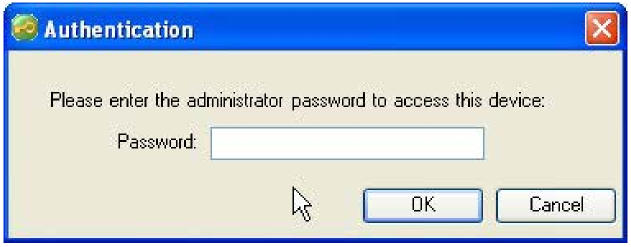
● Get configuration from devices

○ Send configuration to devices

Systems that have authentication enabled will have a “key” symbol visible in the device status indictor as shown above.

Selecting an authenticated system and clicking **Connect** with SoundStructure Studio v1.8 or newer does not cause the Authentication dialog to display, as shown in the following figure.

Authentication Dialog



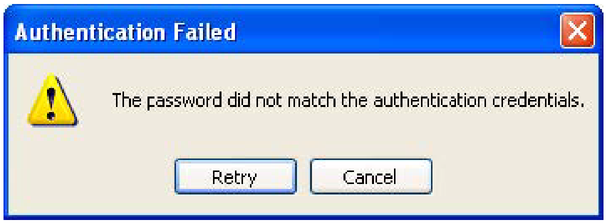
Please enter the administrator password to access this device:

Password:

To connect to the system, enter the system password and click **Ok.**

If the entered password does not match the system password, an error message displays,as shown next.

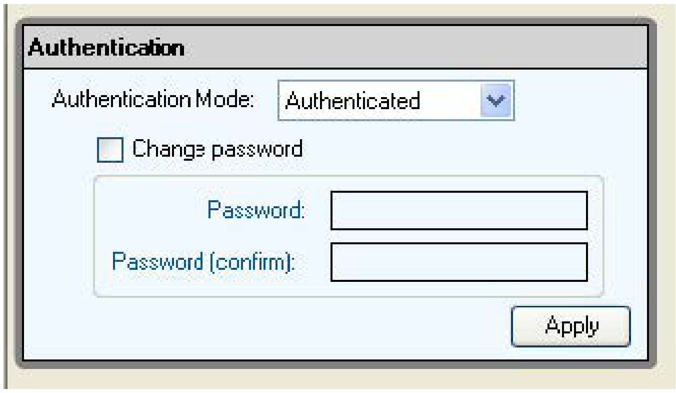
Authentication Failed Dialog



The password did not match the authentication credentials.

If the password is correct, SoundStructure Studio connects to the system and the system displays as Authenticated, as shown in the following figure.

Authenticated SoundStructure System



Authentication Mode: Authenticated

☐ Change password

Password:

Password (confirm):

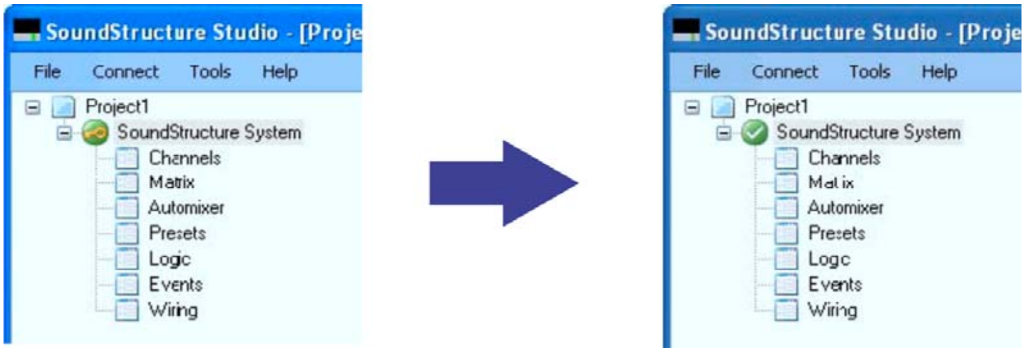
Removing Authentication from a SoundStructure System

To remove authentication from a SoundStructure System, set the Authentication mode to **open** and click **Apply**. SoundStructure Studio will reconnect to the system using port 52774.

The password in an open system will still be retained, but will not be required to connect to the system when it is open mode.

Once the system is set to open, the appearance of the system in SoundStructure Studio will change from the authenticated view on the left side of the following figure to the default open view shown on the right.

Default Open View in SoundStructure Studio


SoundStructure Studio - [Proje
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring
SoundStructure Studio - [Proje
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Logic
Events
Wiring

Viewing SoundStructure Command Logs

SoundStructure system logs show that a system has authentication enabled and will show that an authenticate command has been sent as shown below.

```
Dec 22 17:18:29 gcp: cmd: [1:5:192.168.1.200] authenticate "admin" "****"
Dec 22 17:18:29 gcp: ack: [1:5:192.168.1.200] authenticated
```

Passwords are replaced with “****” in the log and in command acknowledgements to prevent unintended disclosure of the system password.

Understanding SoundStructure System Compatibility Considerations

Using SoundStructure System Control with Third-party Control Systems

Control of a SoundStructure system over RS-232 does not require authentication. The RS-232 interface always operates in an open mode.

For network control, the control system can detect whether authentication is enabled by trying to connect to port 52774. If the session does not connect, the control system should try to connect over port 52775. If the connection succeeds then the SoundStructure system has authentication enabled.

If authentication is enabled, the control system connected to a SoundStructure system over the network at port **52775** can authenticate as follows, assuming the password of the system is 456:

```
authenticate "admin" "456"
```

The SoundStructure device will send back the response:

```
authenticated
```

Once authenticated, SoundStructure API commands can be sent and the associated command acknowledgements will be received.

If the system requires authentication but has not yet been authenticated, any API command sent to the SoundStructure system that is not on the white-list (see below) will respond with:

```
error: "authentication required for control"
```

When authentication is enabled, connections to port 52774 are not allowed.

If authentication is not enabled, then a control system should connect over port 52774 and send commands and receive command acknowledgments over port 52774.

Discovering SoundStructure Devices

When a system is in the open mode, i.e., not authenticated, earlier versions of SoundStructure Studio can discover the SoundStructure system.

When a system is authenticated, earlier versions of SoundStructure Studio will not be able to discover the SoundStructure system.

Supporting SoundStructure Command White-list

SoundStructure system firmware version 1.6 supports a white-list of read-only SoundStructure parameters that may be queried from a SoundStructure system, without authentication, even when the system requires authentication. The white-list commands make it possible to check the status of an authenticated SoundStructure system but not change any settings, without authentication. The white-list commands are desired to allow discovery by SoundStructure Studio and to enabling monitoring applications of SoundStructure systems.

Note: White-List Commands Sent over TCP Port 52775

When authentication is enabled on a SoundStructure system, white-list commands must be sent over TCP port 52775 and the SoundStructure system will respond with command acknowledgments over port 52775. Connections to port 52774 are not supported when authentication is enabled on the SoundStructure system.

The white-list commands are shown in the following table.

White-List of Commands

White-list Parameter Names	
clink_num_attached	dev_status
dev_type	dev_firmware_ver
dev_habanero_ver	dev_hw_eco
dev_hw_rev	dev_plugin_type
dev_uptime	dev_temp
dev_temp_status	dev_volt_clink
dev_volt_neg_15	dev_volt_phantom
dev_volt_pos_15	eth_auth_mode
eth_mac	eth_settings

White-List of Commands

ser_baud	ser_control_mode
ser_flow	sys_bus_id
sys_devices_match	sys_name
sys_num_devices	sys_plugins_match
voip_eth_settings	voip_status
voip_uc_sw_ver	

Understanding SoundStructure Command Sessions

When you send SoundStructure API commands to a system that requires authentication but has not been authenticated, the system responds with the error message:

```
error “authentication required for control”
```

Updating SoundStructure Firmware

Upgrading the SoundStructure device firmware to a newer version will not change the authentication password that has been stored with the SoundStructure System.

Recovering the SoundStructure System Password

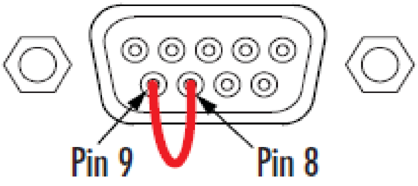
If the SoundStructure system password has been lost, there are two ways to recover.

1. Connect to the system over RS-232 and disable authentication. Once authentication is disabled, you can reconnect to the system over the network port 52774 and change the authentication password.

   Authentication is not applied to the RS-232 interface.
2. Factory reset the device to restore the default open authentication mode with the default password of 456. Please note this will erase the settings from the SoundStructure system.

To factory reset a SoundStructure system that you are not able to connect to, wire pins 8 and 9 of the RS-232 interface together as shown in the following figure and power cycle the SoundStructure device. After the system starts rebooting and the front panel LED flashes green, remove the connection between pins 8 and 9.

Pin 8 and Pin 9



After the system has finished booting, the system is in a factory reset state with no configuration file settings.

Note: Factory Resetting Erases all System Settings

Factory resetting a SoundStructure system will erase all system settings in the device including the configuration file, all SoundStructure VoIP interface settings, and will reset the SoundStructure system password to the default of 456 and set the operating mode to ‘open’.

To save your SoundStructure VoIP Interface settings, power down the SoundStructure system and remove the SoundStructure VoIP Interface before factory resetting the SoundStructure system.

SoundStructure Authentication API Command Summary

New SoundStructure API parameters and a new SoundStructure command action were added to support authentication.

Authenticate

The authenticate command action allows a user to authenticate to a SoundStructure system. The command syntax is:

authenticate “username” “password”

where username should be set to “admin” and the password must match the password configured for the system. By default the password is set to 456 and a default system could be authenticated with the command:

```
authenticate “admin” “456”
```

The system would respond with the acknowledgment

```
authenticated
```

If the password is not correct, the system will respond with:

```
error “authentication failed”
```

Understanding SoundStructure Authentication Parameters

These new parameters are shown in the following table.

Authentication Parameters

Parameter Name	Description
auth_password	Used to set the password for the system. The password must be less than 128 characters. Note that the password is sent as clear text. There is no encryption associated with transmitting the password to a SoundStructure system.
eth_auth_mode	Used to set the authentication mode to either open or auth. When set to open, TCP communication occurs over port 52774 and TCP communication over port 52775 is disabled. When set to auth, TCP communication occurs over port 52775 and TCP communication over port 52774 is disabled.

Authentication Parameters

sys_num_auth_connections	This parameter returns the total number of Ethernet connections for which `eth_auth_mode` is set to auth.
sys_num_connections	This parameter returns the total number of Ethernet connections to the system and is equal to the sum of s`ys_num_auth_connections` + `sys_num_open_connections`.
sys_num_open_connections	This parameter returns the total number of Ethernet connections for which `eth_auth_mode` is open.

Creating Advanced Applications

This chapter describes several applications of the SoundStructure products and the steps required to create these applications. These applications include the following conferencing applications:

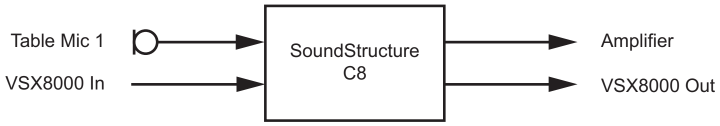
- Creating a One Microphone And Mono Video Conferencing System
- Creating Four Digital Array Microphones and A SoundStation VTX1000 Conferencing System
- Creating an Eight Microphones, Video, and Telephony Application Conferencing System
- Creating a Two PSTN Line Positional “Receive” Audio Conferencing System
- Creating an Eight Microphones and Stereo Video Conferencing System
- Creating an Eight Microphones with The Polycom Video Codec Conferencing System
- Creating an Eight Microphones with Wireless and Lectern Microphones Reinforcement Conferencing System
- Creating a Sixteen Microphones with Six-Zone Sound Reinforcement Conferencing System
- Creating a Room Combining Application Conferencing System

Creating a One Microphone And Mono Video Conferencing System

This simple example is designed to show how to get started designing with the SoundStructure products. In this example, one microphone and a Polycom VSX8000 are used with a SoundStructure C8 device.

The block diagram of this system is shown in the following figure. The channel names are labeled with the virtual channel names that are created by default by the SoundStructure Studio software.

Block Diagram for One Microphone and Polycom VSX8000 with SoundStructure C8

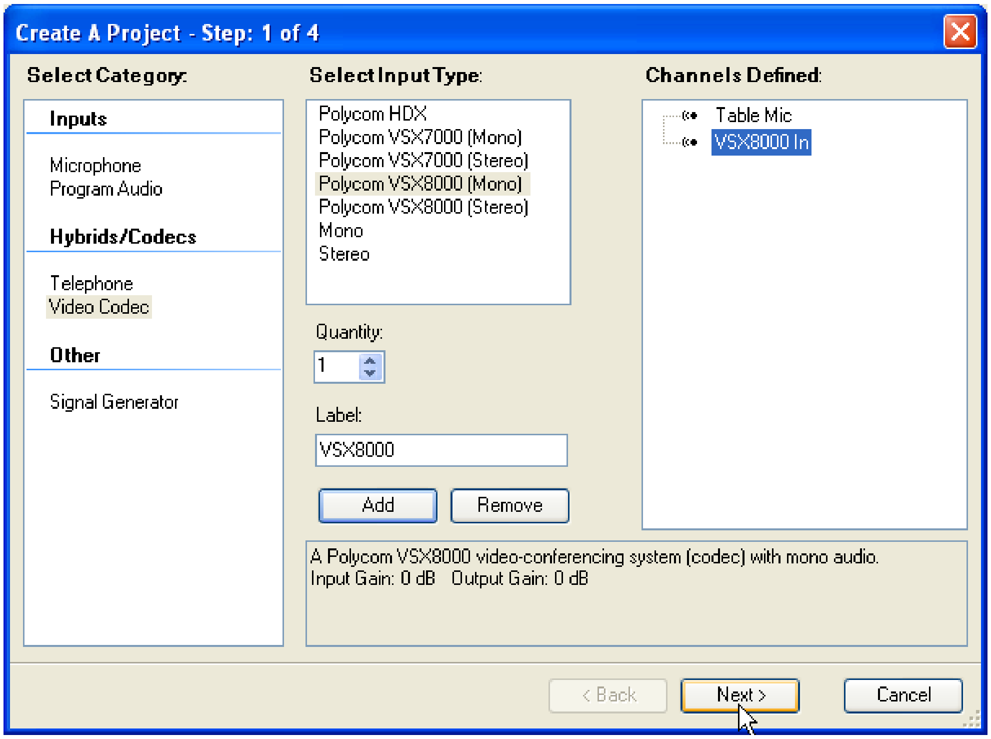


Before proceeding with the design, install SoundStructure Studio software from the CD-ROM supplied with your SoundStructure device or download the latest version from the Polycom website. Launch the SoundStructure Studio software and select New Project from the File menu.

SoundStructure Studio Steps

Step 1 - Select Inputs

For the first step, select one table top microphone and a VSX8000 mono video codec.


Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Telephone
Video Codec
Other
Signal Generator
Select Input Type:
Polycom HDX
Polycom VSX7000 (Mono)
Polycom VSX7000 (Stereo)
Polycom VSX8000 (Mono)
Polycom VSX8000 (Stereo)
Mono
Stereo
Add
Remove
Channels Defined:
Table Mic
VSX8000 In
< Back
Next >
Cancel

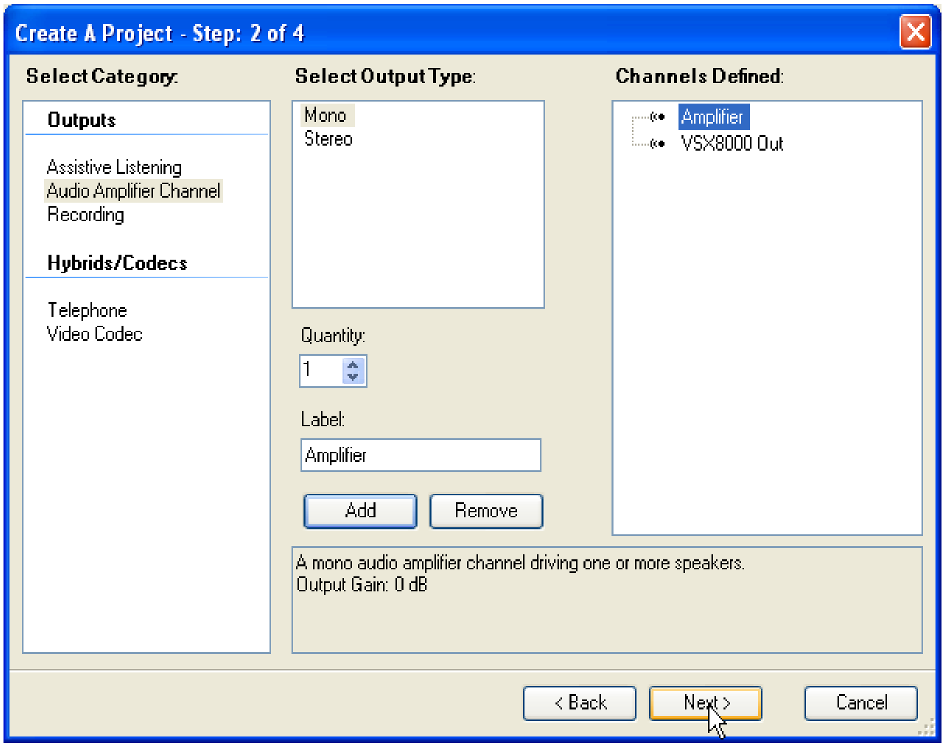
Quantity: 1

Label: VSX8000

A Polycom VSX8000 video-conferencing system (codec) with mono audio.
Input Gain: 0 dB Output Gain: 0 dB

Step 2 - Select Outputs

For the second step, select a mono amplifier as the output source. The VSX8000 output is automatically defined when the VSX8000 input is selected.


Create A Project - Step: 2 of 4
Select Category:
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Telephone
Video Codec
Select Output Type:
Mono
Stereo
Channels Defined:
Amplifier
VSX8000 Out
Add
Remove
< Back
Next >
Cancel

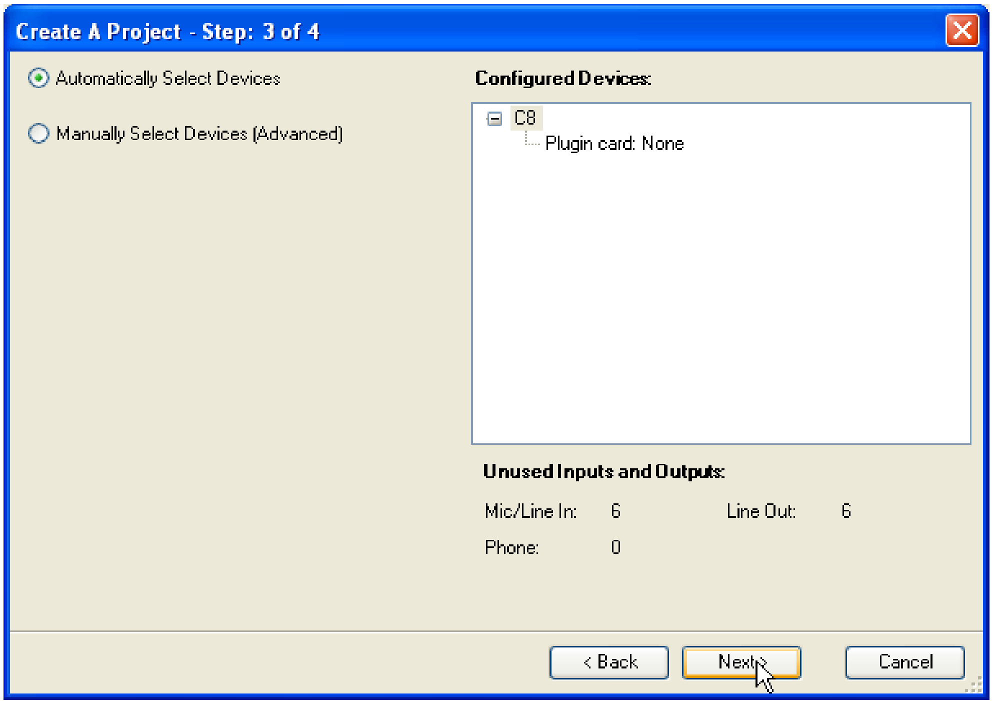
Quantity: 1

Label: Amplifier

A mono audio amplifier channel driving one or more speakers.
Output Gain: 0 dB

Step 3 - Select Equipment

Select the equipment required to create this design. By default a SoundStructure C8 is selected.


Create A Project - Step: 3 of 4
< Back
Next >
Cancel

● Automatically Select Devices

○ Manually Select Devices (Advanced)

Configured Devices:

- C8
  - Plugin card: None

Unused Inputs and Outputs:

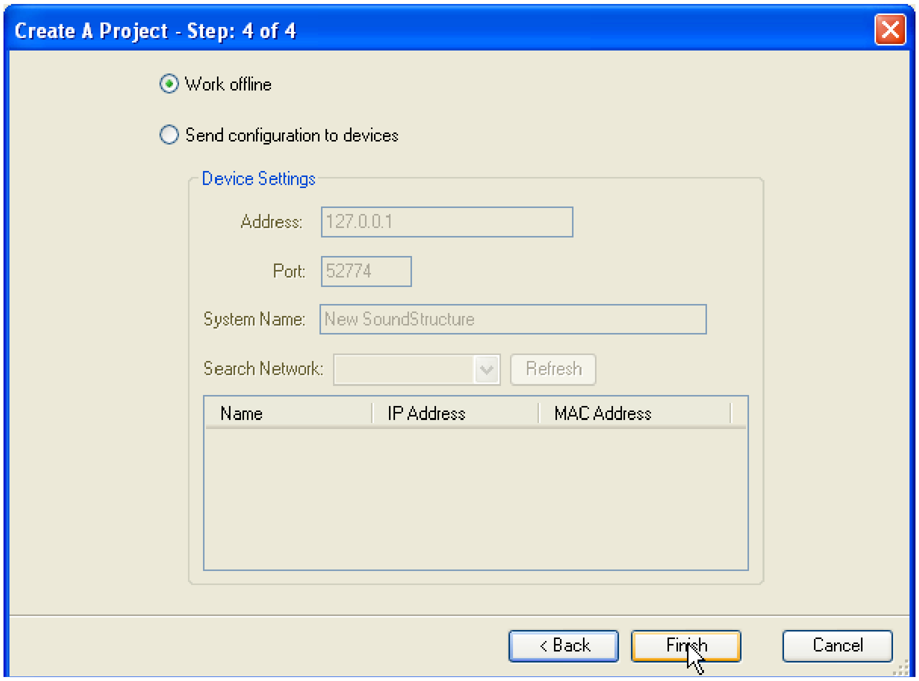
Mic/Line In: 6

Line Out: 6

Phone: 0

Step 4 - Work Offline Or Online

In this step offline operation is selected to create a file for later upload into a SoundStructure C8.



Create A Project - Step: 4 of 4

● Work offline

○ Send configuration to devices

Device Settings

Address: 127.0.0.1

Port: 52774

System Name: New SoundStructure

Search Network:

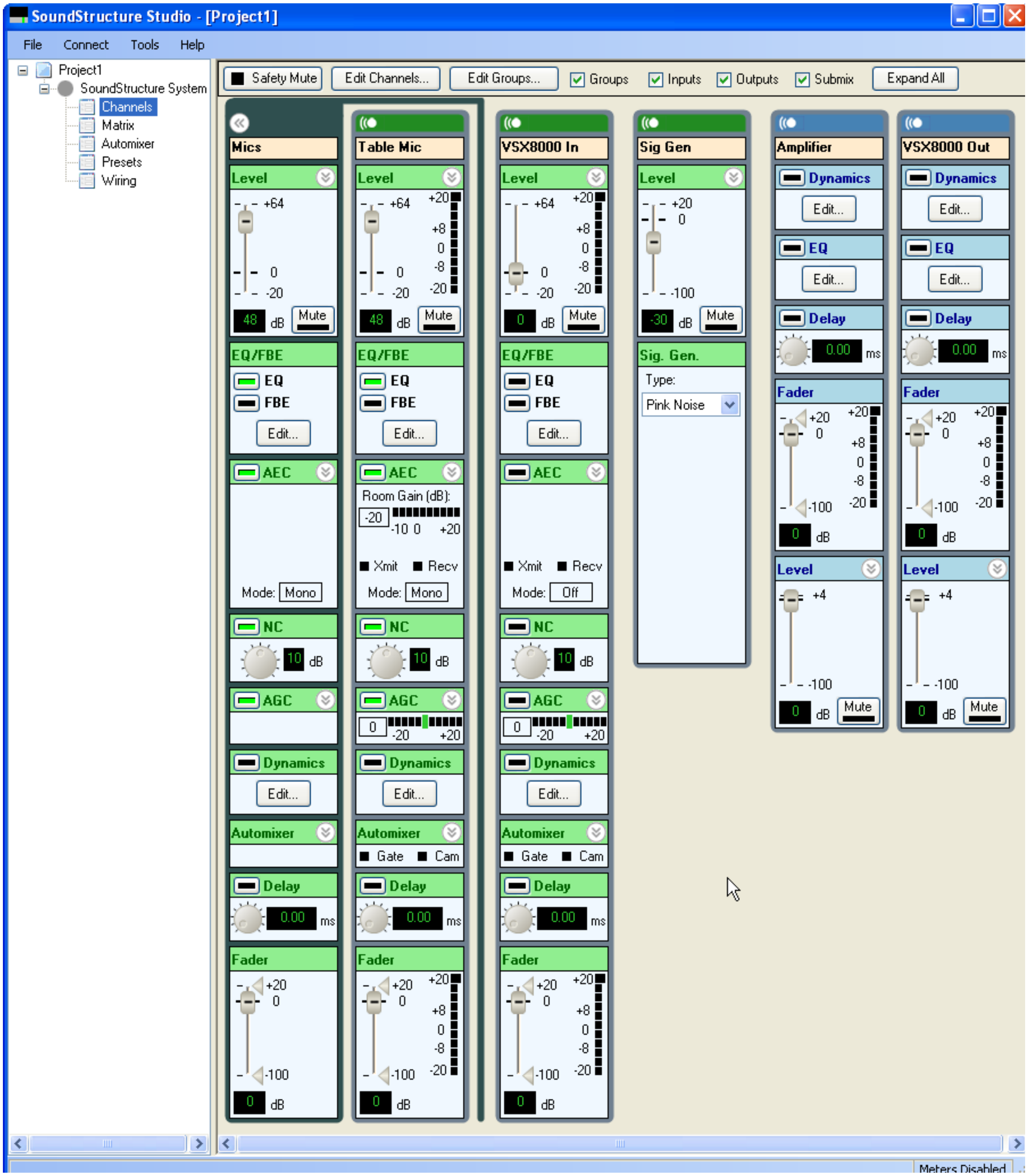
Using the Channels Page

After you have created the design, the Channels page displays where the following virtual channels and virtual channel groups have been defined.

Virtual Channels and Descriptions

Channel	Description
Mics	A virtual channel group that includes Table Mic 1
Table Mic 1	A table top microphone with phantom power enabled and a default gain of 48dB
VSX8000 In	The audio output from the VSX8000 that is an input to the SoundStructure device
Amplifier	The output to the amplifier that will drive audio into the local room
VSX8000 Out	The audio output from the SoundStructure device that is an input to the video codec
Sig Gen	A signal generator that can be used for setting amplifier volume levels and checking that loudspeakers are connected.

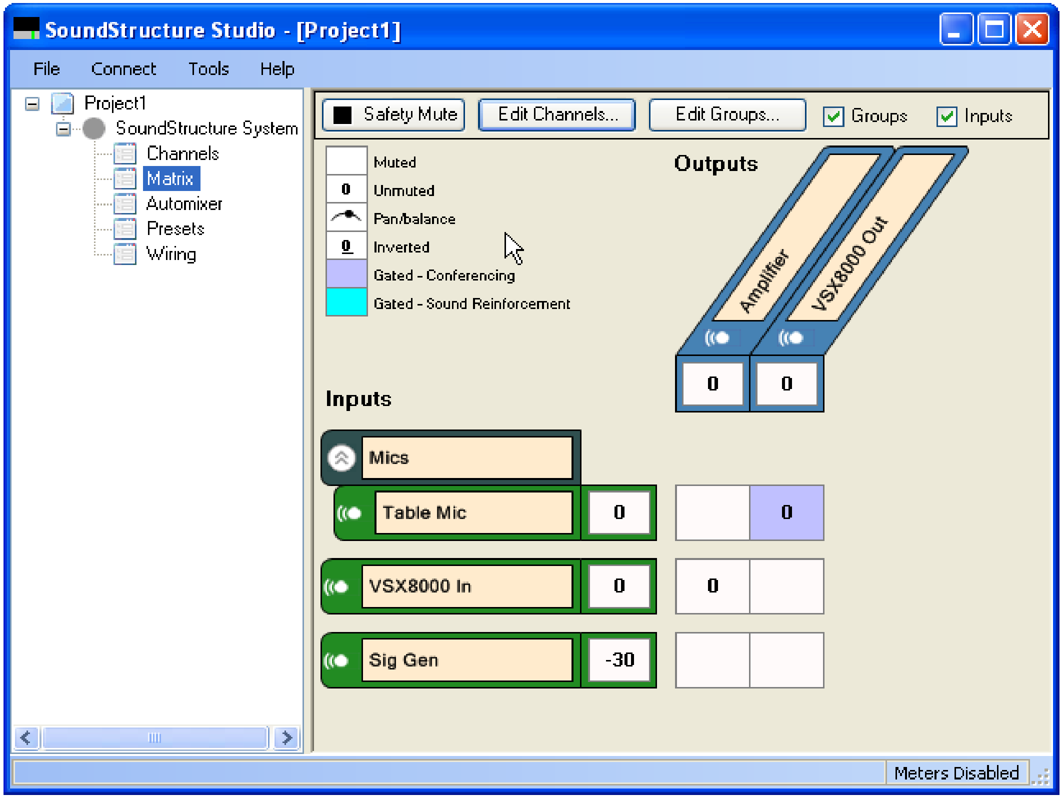
These channels are shown in the channels page in the following figure. The input gain for tabletop microphone is set to 48dB. Since the VSX8000 has a 0 dBu nominal input and output signal, the input gain for the VSX8000 In channel is set to 0dB, in other words, no gain is applied. It is also assumed that the Amplifier can accept the nominal 0dBu level from the SoundStructure device, allowing the SoundStructure Amplifier output to have 0dB output gain. If the Amplifier input has an RCA connection, the Amplifier output gain adjusted from 0dB to -10dB to prevent overdriving the consumer-level input on the Amplifier.


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Edit Groups...
Groups
Inputs
Outputs
Submix
Expand All
Mics
Table Mic
VSX8000 In
Sig Gen
Amplifier
VSX8000 Out
Level
EQ/FBE
EQ
FBE
AEC
Room Gain (dB):
Xmit
Recv
Mode: Mono
Mode: Off
NC
AGC
Dynamics
Automixer
Gate
Cam
Delay
Fader
Sig. Gen.
Type:
Pink Noise
Mute
Edit...
Meters Disabled

Using the Matrix Page

The matrix page shows how the input signals are mapped to the output signals. In this example, the tabletop microphone is sent to the VSX8000 and the VSX8000 is sent to the local amplifier. The signal generator is muted.

Matrix Page

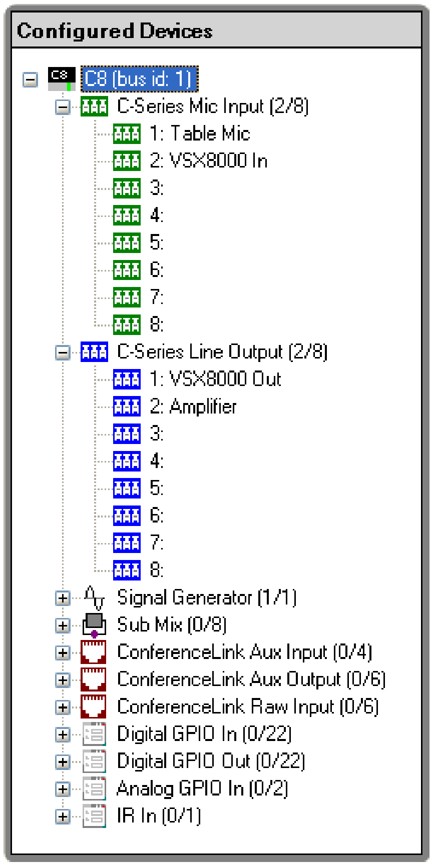

SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Edit Groups...
Groups
Inputs
Muted
Unmuted
Pan/balance
Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Amplifier
VSX8000 Out
Inputs
Mics
Table Mic
VSX8000 In
Sig Gen
-30
Meters Disabled

Understanding Wiring Information

The system should be cabled according to the layout on the wiring page as shown in the following figure. To wire the system with virtual channels on different physical inputs or outputs, drag the channels to their desired physical inputs or outputs and then cable the system according to the updated wiring information.

In this example, Table Mic 1 is connected to physical input 1, the VSX8000 In channel is connected to physical input 2, the VSX8000 Out channel is connected to physical output 1 and the Amplifier channel is connected to physical output 2. If this wiring scheme does not match how the system has been wired, the channels may be moved around on the wiring page to their desired locations.

Wiring Information



Configured Devices

- C8 (bus id: 1)
  - C-Series Mic Input (2/8)
    - 1: Table Mic
    - 2: VSX8000 In
    - 3:
    - 4:
    - 5:
    - 6:
    - 7:
    - 8:
  - C-Series Line Output (2/8)
    - 1: VSX8000 Out
    - 2: Amplifier
    - 3:
    - 4:
    - 5:
    - 6:
    - 7:
    - 8:
  - Signal Generator (1/1)
  - Sub Mix (0/8)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

Controlling The System

A control system will typically be used to mute the microphone and adjust the volume settings in the local room. The following sections describe how this may be done with the command syntax of the SoundStructure devices. See Appendix A - Command Protocol Reference Guide for additional information on the command set.

Using the Mute Controls

The microphones in the system may be muted either individually or as the “Mics” group by sending the following API command to the SoundStructure device:

```
set mute “Mics” 1
```

will mute all the microphone in the system and

```
set mute “Mics” 0
```

will unmute the microphone in the system.

Using the Volume Controls

Volume control in the room can be accomplished by adjusting the fader control on the “Amplifier” virtual channel as follows:

```
inc fader “Amplifier” 1
```

will increase the gain on the “Amplifier” channel by 1dB and

```
dec fader “Amplifier” 1
```

Alternatively the fader settings may be set to an absolute value with the set command as follows:

```
set fader “Amplifier” 0
```

to set the value of the fader to 0dB.

The volume control range can be limited by setting a fader max and fader min as shown in the API syntax below:

```
set fader max “Amplifier” 10
set fader min “Amplifier” -10
```

to limit the maximum and minimum user range of the fader control to +10 and -10dB respectively. The max and min ranges only need to be set once and can be configured as part of the SoundStructure Studio configuration file. If the current amplifier fader setting is outside of this range, the range of the maximum or minimum fader values will be adjusted to include the current fader setting.

In other words, to set a fader max or min value, the current fader value must be within the range of values. Otherwise the range is extended to include the current fader value.

Creating Four Digital Array Microphones and A SoundStation VTX1000 Conferencing System

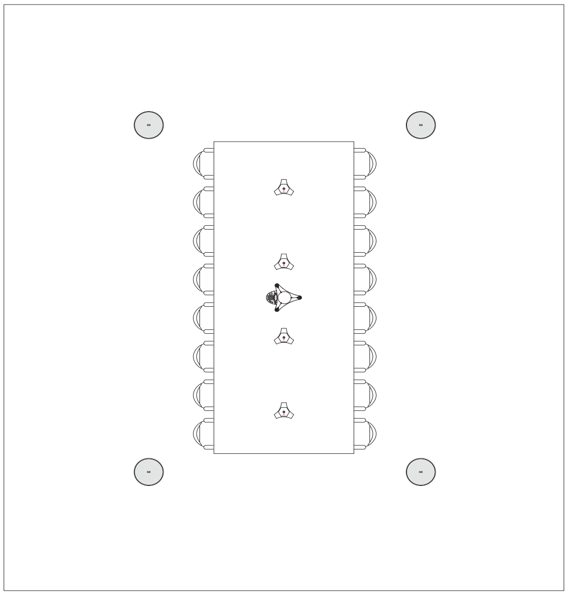
This example creates a typical audio conferencing system with four digital microphone arrays, mono program audio, a SoundStation VTX1000, and a single audio amplifier zone. In this application the VTX1000 will be the analog telephony interface and can be used to make telephone calls and to control volume in the local room with the volume adjustment on the VTX1000. The system operates as follows:

- This VTX1000 volume control will adjust the level of the phone line signal that is an input to the SoundStructure device and increase level in the local room.
- The VTX1000 mute button will mute the audio that is transmitted down the VXT1000’s telephone line so the remote telephony participants won’t be able to hear the local participants while muted.
- The VTX1000 must be configured for “Vortex” mode to route the appropriate signals to and from the Aux In and Aux Out connectors on the VTX1000 power supply.
- The VTX1000’s microphones and loudspeaker are not used in this configuration.

Digital microphones are used in this example for ease of installation, however traditional analog microphones could also be used in the system.

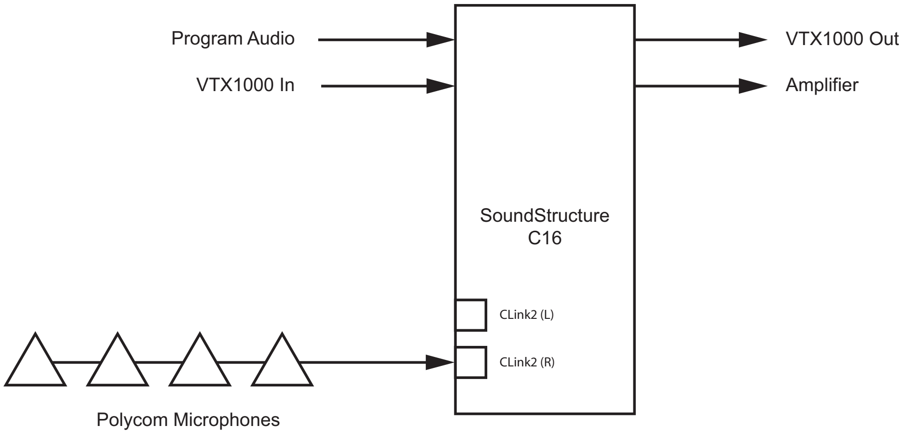
The room layout may look similar to the reflected ceiling plan shown in the following figure with in-ceiling loudspeakers, a SoundStation VTX1000 on the front of the room, and the digital microphone arrays distributed on the table.

Room Layout for Conferencing System



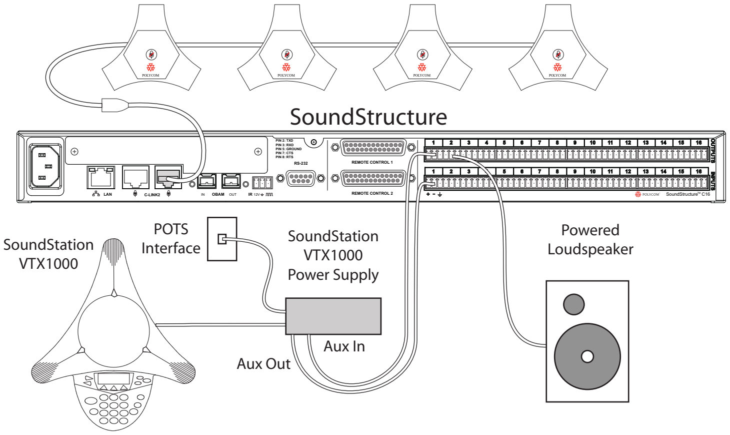
The block diagram of this system is shown in the following figure.

Block Diagram for Conferencing System


Program Audio
VTX1000 In
VTX1000 Out
Amplifier
SoundStructure
C16
CLink2 (L)
CLink2 (R)
Polycom Microphones

The From VTX1000 and To VTX10000 signals are wired to the VTX1000 power module as shown in the following figure.

VTX1000 Signals Wired to VTX1000 Power Module

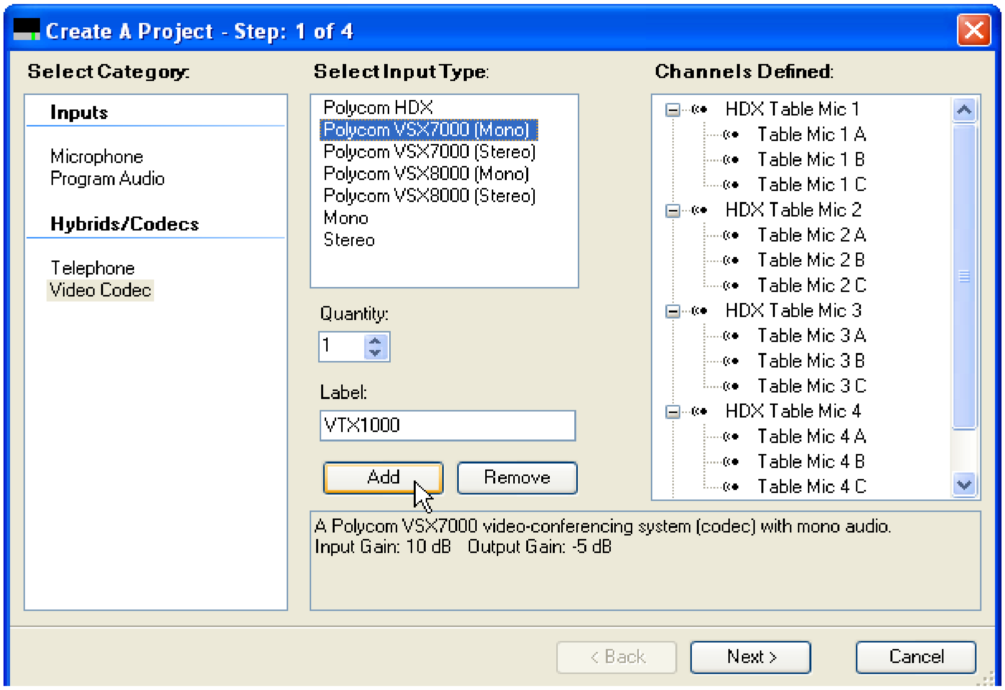

SoundStructure
SoundStation VTX1000
POTS Interface
SoundStation VTX1000 Power Supply
Powered Loudspeaker
Aux Out
Aux In

SoundStructure Studio Steps

The steps to create this project are shown in the following figures. The names for the channels are the names that SoundStructure Studio defines.

Step 1 - Select Inputs

Select four HDX digital tabletop microphones and a mono program audio source. If the VTX1000 isn't listed, select the VSX7000 video conferencing system and adjust the labels as shown in the following figure.



Create A Project - Step: 1 of 4

Select Category:

Inputs
- Microphone
- Program Audio

Hybrids/Codecs
- Telephone
- Video Codec

Select Input Type:
- Polycom HDX
- Polycom VSX7000 (Mono)
- Polycom VSX7000 (Stereo)
- Polycom VSX8000 (Mono)
- Polycom VSX8000 (Stereo)
- Mono
- Stereo

Quantity: 1

Label: VTX1000

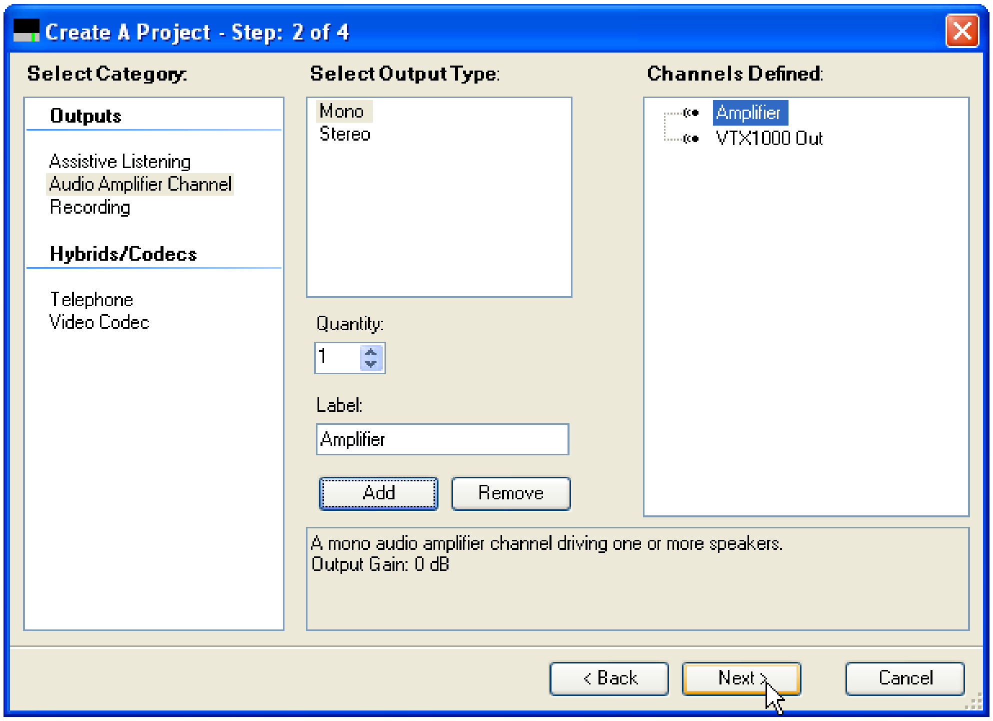
Channels Defined:
- HDX Table Mic 1
  - Table Mic 1 A
  - Table Mic 1 B
  - Table Mic 1 C
- HDX Table Mic 2
  - Table Mic 2 A
  - Table Mic 2 B
  - Table Mic 2 C
- HDX Table Mic 3
  - Table Mic 3 A
  - Table Mic 3 B
  - Table Mic 3 C
- HDX Table Mic 4
  - Table Mic 4 A
  - Table Mic 4 B
  - Table Mic 4 C

A Polycom VSX7000 video-conferencing system (codec) with mono audio.
Input Gain: 10 dB Output Gain: -5 dB

Add
Remove
< Back
Next >
Cancel

Step 2 - Select Outputs

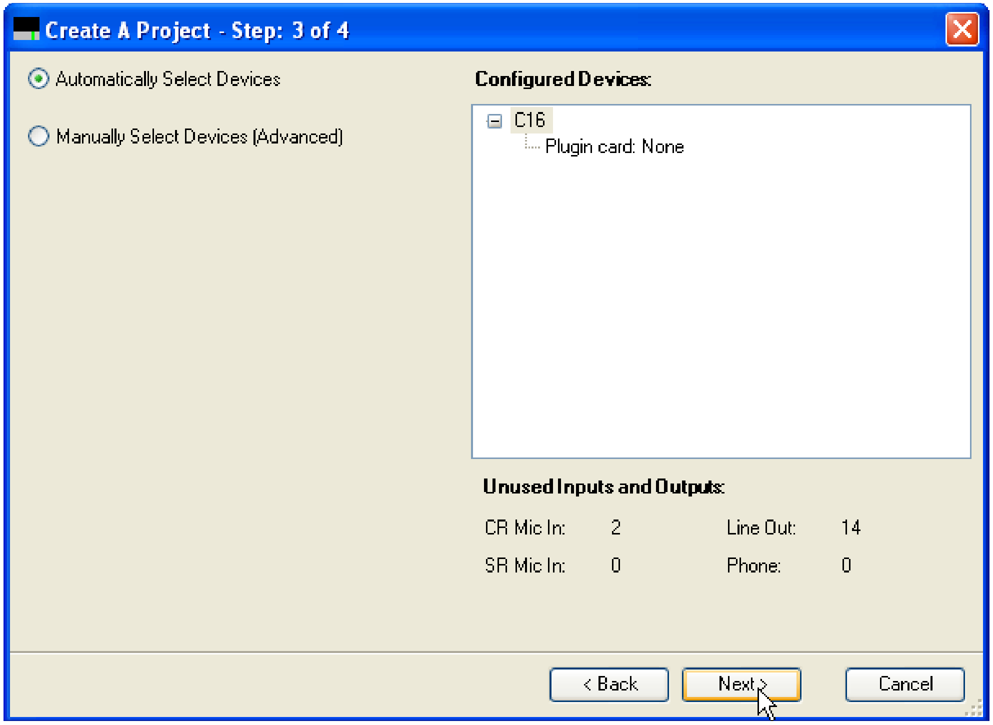
Select a mono amplifier as the output source. The VTX1000 output will be automatically defined when the VTX1000 input is defined.


Create A Project - Step: 2 of 4
Select Category:
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Telephone
Video Codec
Select Output Type:
Mono
Stereo
Quantity:
1
Label:
Amplifier
Add
Remove
Channels Defined:
Amplifier
VTX1000 Out
< Back
Next >
Cancel

A mono audio amplifier channel driving one or more speakers.
Output Gain: 0 dB

Step 3 - Select Devices

Select the equipment required to create this design. By default the SoundStructure C16 is selected. Note that no telephony card is required as the VTX1000 will be the telephony interface.



● Automatically Select Devices

○ Manually Select Devices (Advanced)

Configured Devices:

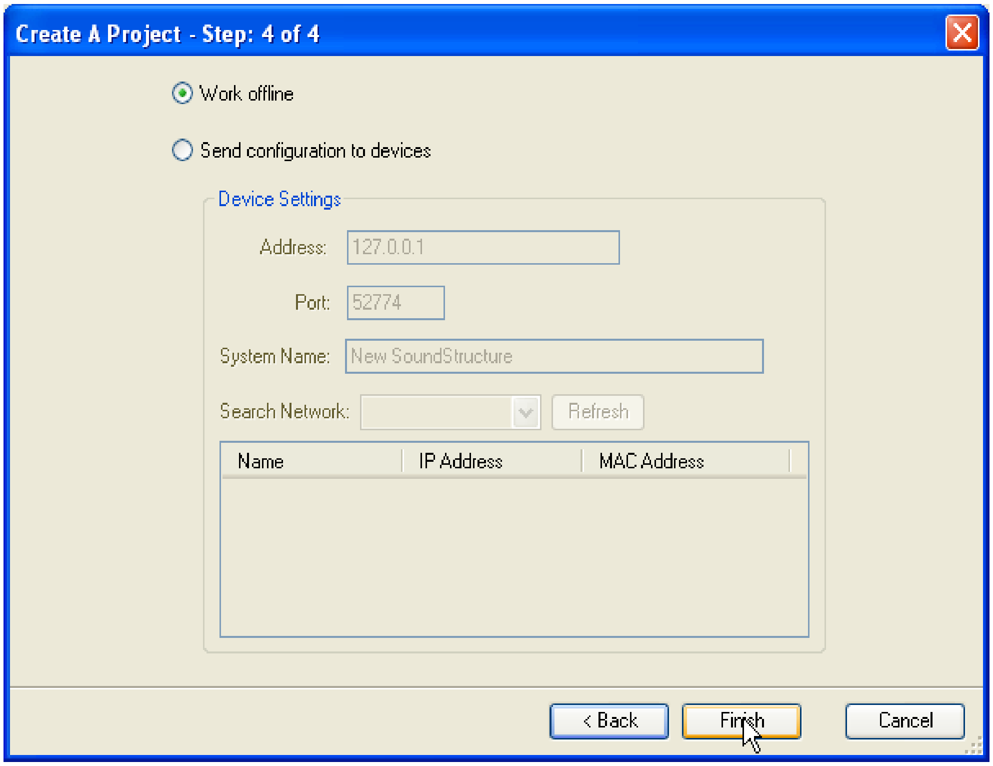
- C16
  - Plugin card: None

Unused Inputs and Outputs:

CR Mic In:	2	Line Out:	14
SR Mic In:	0	Phone:	0

Step 4 - Work Offline Or Online

In this step offline operation is selected to create a file for later upload into a SoundStructure C16.



Create A Project - Step: 4 of 4

● Work offline

○ Send configuration to devices

Device Settings

Address: 127.0.0.1

Port: 52774

System Name: New SoundStructure

Search Network:

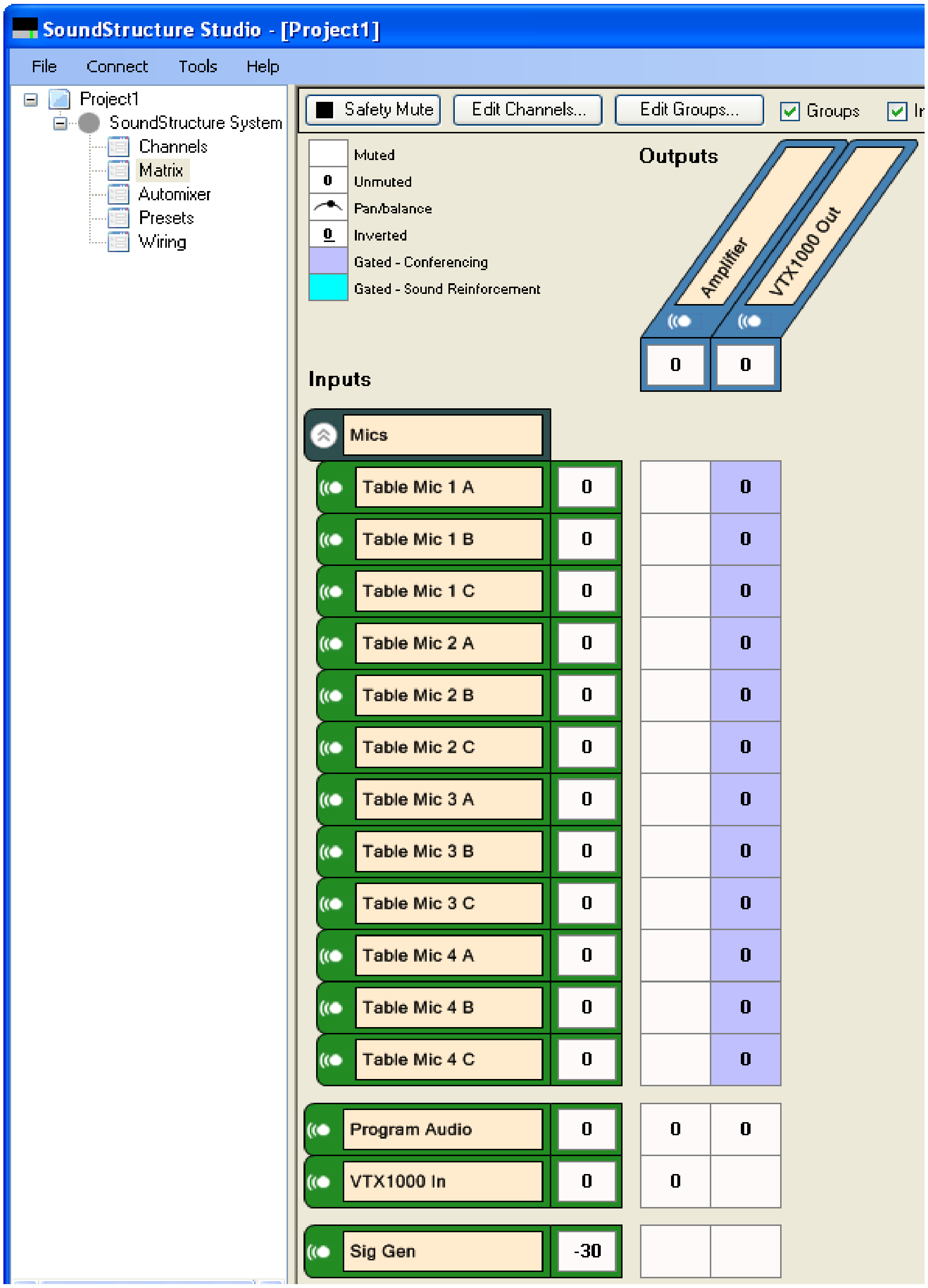
Editing Matrix Settings

After you have designed the system, click the Matrix label in the project window to view the matrix shown in the following figure.

The input virtual channels that include remote audio are the “VTX1000 In” and “Program Audio”. These channels are routed to the “Amplifier” channel so they can be heard in the local room.

The microphones “Table Mic 1 A” through “Table Mic 4 C” are routed to the “VTX1000 Out” channel using the conferencing signal path which includes echo and noise cancellation, and automixer processing. The blue background of these crosspoints is the visual indicator that the conferencing version of the input processing has been selected.

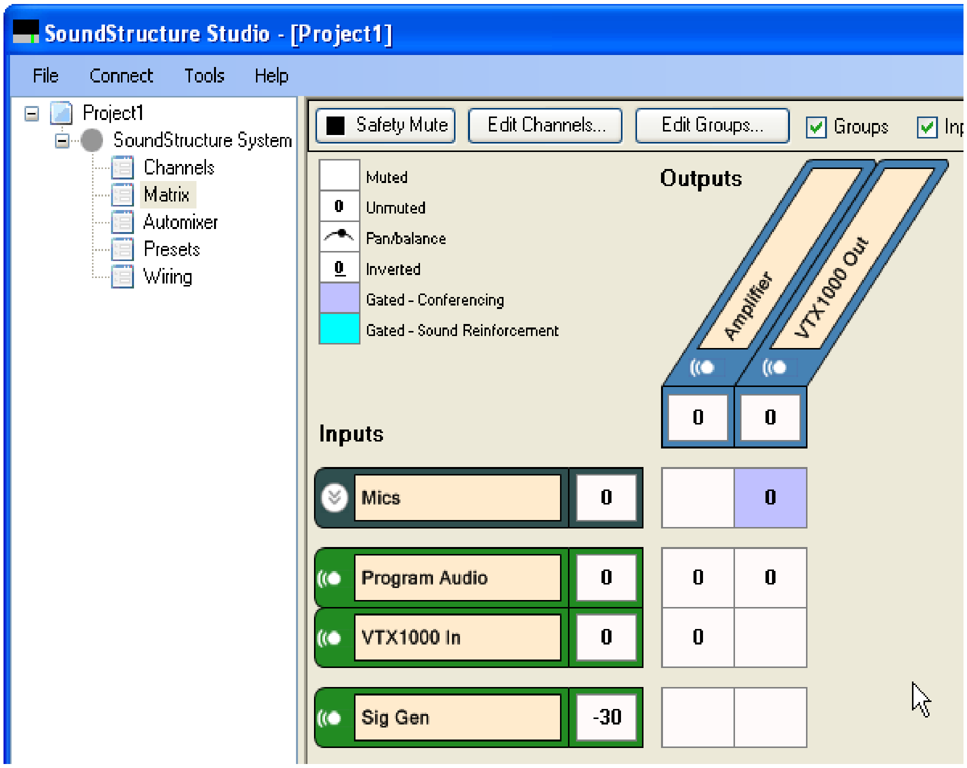
Matrix Label in Project Window


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Ir
Muted
Unmuted
Pan/balance
Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Amplifier
VTX1000 Out
Inputs
Mics

Inputs	Gain	Amplifier	VTX1000 Out
Outputs		0	0
Table Mic 1 A	0		0
Table Mic 1 B	0		0
Table Mic 1 C	0		0
Table Mic 2 A	0		0
Table Mic 2 B	0		0
Table Mic 2 C	0		0
Table Mic 3 A	0		0
Table Mic 3 B	0		0
Table Mic 3 C	0		0
Table Mic 4 A	0		0
Table Mic 4 B	0		0
Table Mic 4 C	0		0
Program Audio	0	0	0
VTX1000 In	0	0	
Sig Gen	-30		

The matrix may be collapsed by clicking the up arrows next to the “Mics” group. Because all the microphones are used in the same way, the group crosspoint represents how all the table microphone channels are being used. The result is a compact matrix representation as shown in the following figure.

Collapsed View of Matrix Page

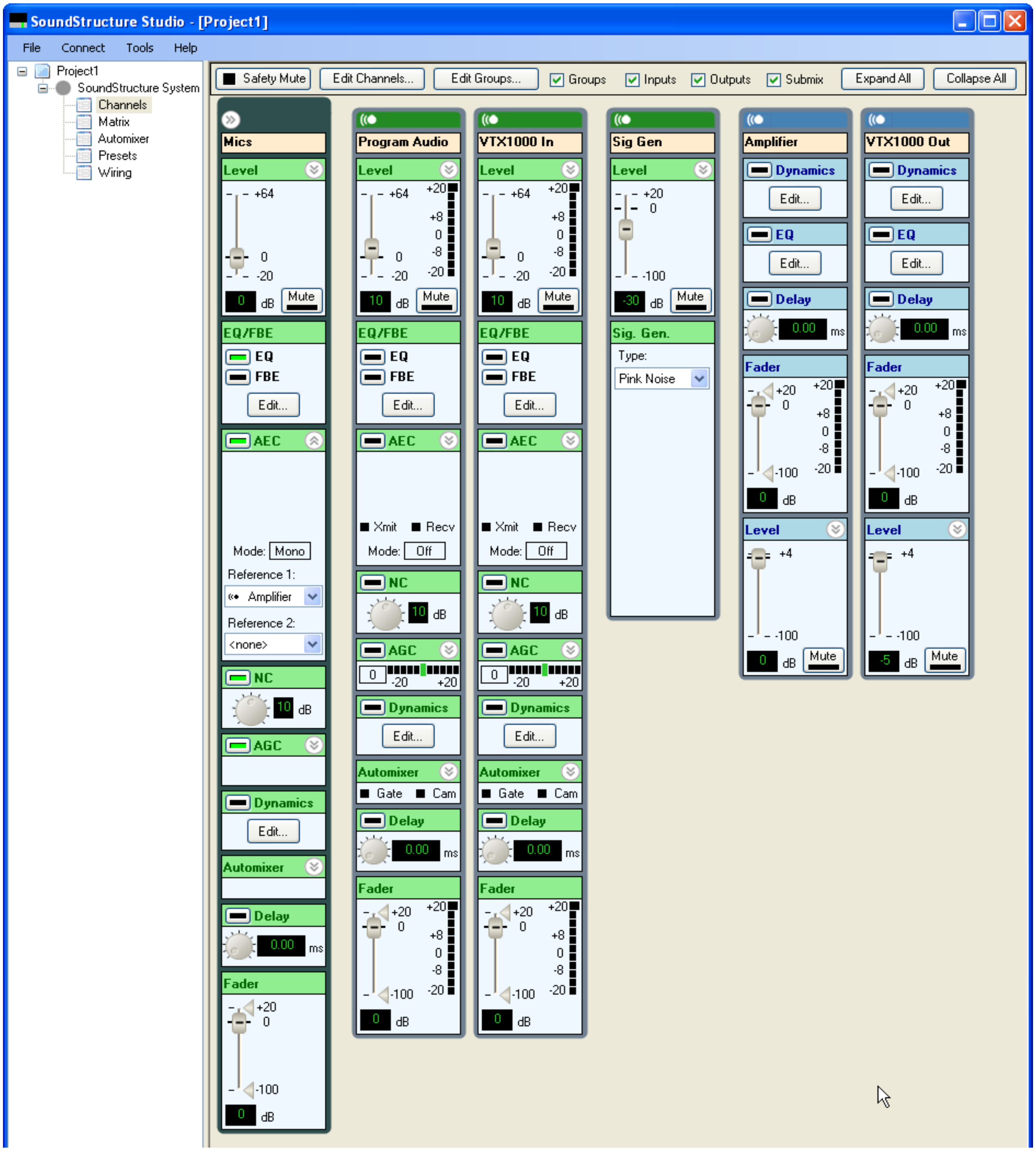

SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inp
Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Amplifier
VTX1000 Out
0
0
Inputs
Mics
0
0
Program Audio
0
0
0
VTX1000 In
0
0
Sig Gen
-30

Editing Channels Settings

The channels page associated with this matrix is shown in the following figure. If the channels are collapsed in the matrix, they are also collapsed in the channels page. The AEC block has been expanded to show the AEC reference.

By default the AEC reference has been set to the mono virtual channel “Amplifier” because this audio includes all the remote audio that need to be echo canceled.

Channels Page Associated with Matrix


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Edit Groups...
Groups
Inputs
Outputs
Submix
Expand All
Collapse All
Mics
Program Audio
VTX1000 In
Sig Gen
Amplifier
VTX1000 Out
Level
EQ/FBE
EQ
FBE
Edit...
AEC
Xmit
Recv
NC
AGC
Dynamics
Automixer
Gate
Cam
Delay
Fader
Sig. Gen.
Mute
dB
ms

Mode: Mono

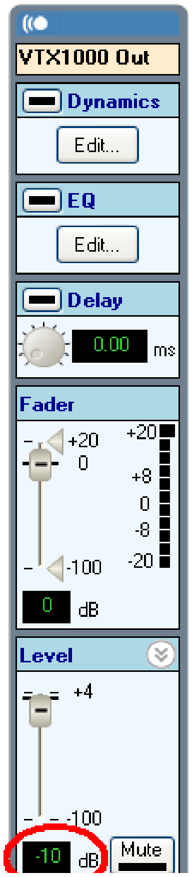
Reference 1: Amplifier

Reference 2: <none>

Type: Pink Noise

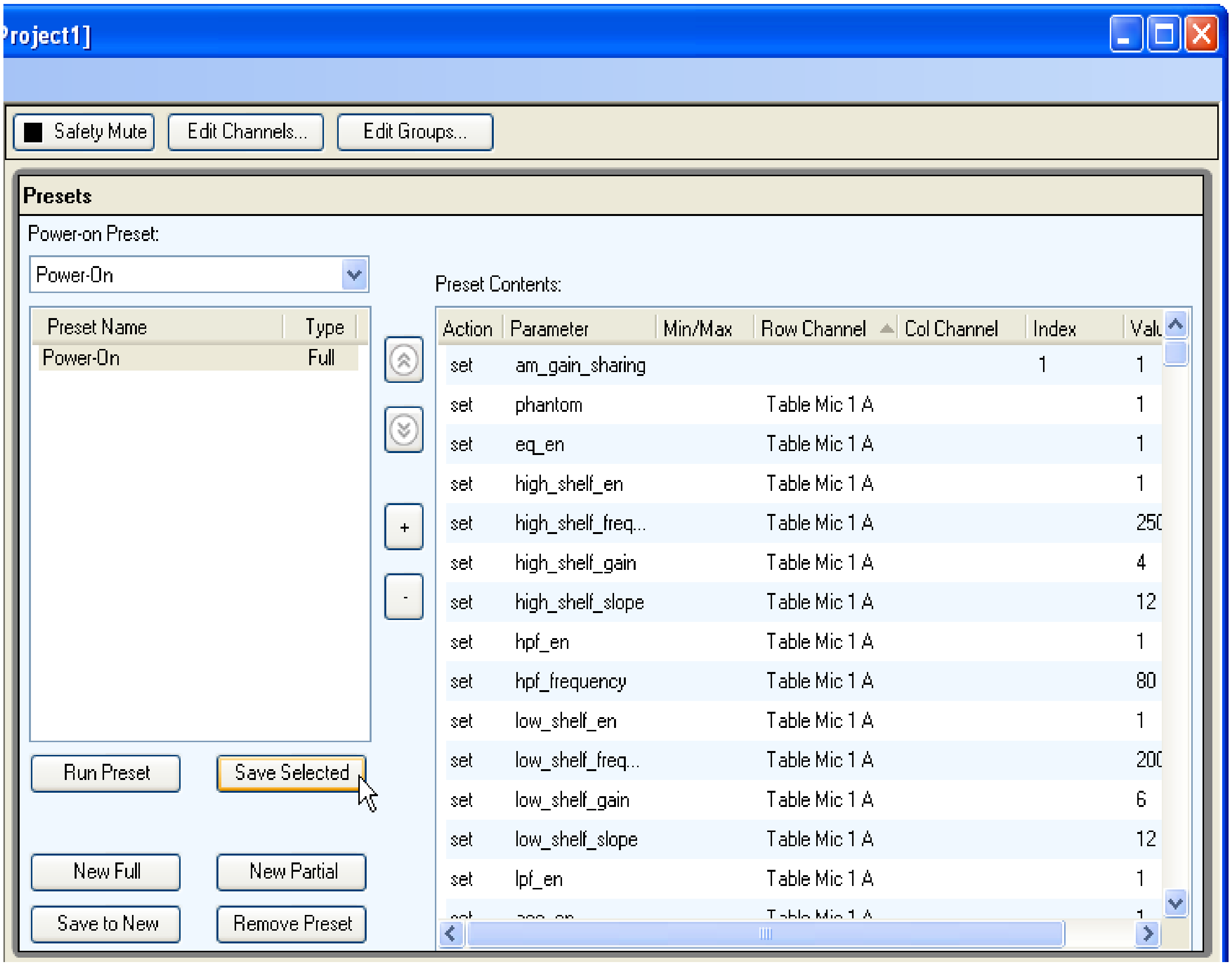
On the VTX1000 out channel, change the output gain from -5 to -10 as shown in the following figure. This change is to ensure the SoundStructure’s output signals at 0du do not overdrive the input of the VTX1000 which is expecting a -10dBu nominal signal.

Output Gain on VTX1000 Out Channel


VTX1000 Out
Dynamics
Edit...
EQ
Edit...
Delay
0.00
ms
Fader
+20
0
-100
+20
+8
0
-8
-20
0
dB
Level
+4
-100
-10
dB
Mute

After this output gain change, and any other changes that are made to the file, the next step is to save the settings to the power on preset as shown on the presets page and in the following figure to ensure all changes are stored permanently inside the system.

Saving Power Preset Settings on Presets Page


roject1]
Safety Mute
Edit Channels...
Edit Groups...
Presets
Run Preset
Save Selected
New Full
New Partial
Save to New
Remove Preset
+
-

Power-on Preset: Power-On

Preset Name	Type
Power-On	Full

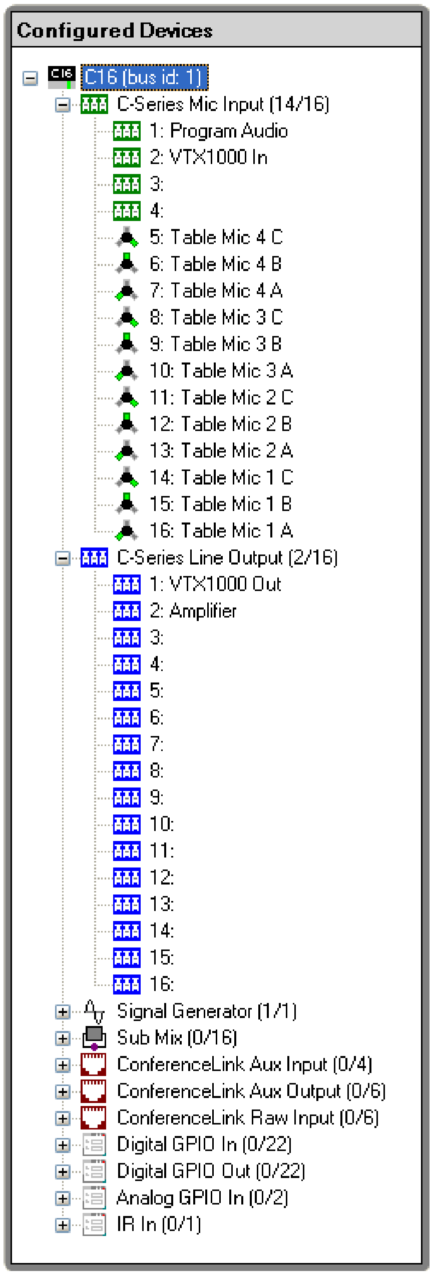
Preset Contents:

Action	Parameter	Min/Max	Row Channel	Col Channel	Index	Valu
set	am_gain_sharing				1	1
set	phantom		Table Mic 1 A			1
set	eq_en		Table Mic 1 A			1
set	high_shelf_en		Table Mic 1 A			1
set	high_shelf_freq...		Table Mic 1 A			25
set	high_shelf_gain		Table Mic 1 A			4
set	high_shelf_slope		Table Mic 1 A			12
set	hpf_en		Table Mic 1 A			1
set	hpf_frequency		Table Mic 1 A			80
set	low_shelf_en		Table Mic 1 A			1
set	low_shelf_freq...		Table Mic 1 A			20
set	low_shelf_gain		Table Mic 1 A			6
set	low_shelf_slope		Table Mic 1 A			12
set	lpf_en		Table Mic 1 A			1

Understanding Wiring Information

The system should be wired according to the layout on the wiring page as shown in the following figure. To wire the system with virtual channels on different physical inputs or outputs, simply drag the channels to their desired locations and then wire the system according to the updated wiring information. The digital microphone arrays require the processing of 12 analog inputs and are assigned to inputs 5 - 16 automatically, leaving the first four analog inputs available to be used with analog signals.

Wiring Information



Configured Devices

- C16 (bus id: 1)
  - C-Series Mic Input (14/16)
    - 1: Program Audio
    - 2: VTX1000 In
    - 3:
    - 4:
    - 5: Table Mic 4 C
    - 6: Table Mic 4 B
    - 7: Table Mic 4 A
    - 8: Table Mic 3 C
    - 9: Table Mic 3 B
    - 10: Table Mic 3 A
    - 11: Table Mic 2 C
    - 12: Table Mic 2 B
    - 13: Table Mic 2 A
    - 14: Table Mic 1 C
    - 15: Table Mic 1 B
    - 16: Table Mic 1 A
  - C-Series Line Output (2/16)
    - 1: VTX1000 Out
    - 2: Amplifier
    - 3:
    - 4:
    - 5:
    - 6:
    - 7:
    - 8:
    - 9:
    - 10:
    - 11:
    - 12:
    - 13:
    - 14:
    - 15:
    - 16:
  - Signal Generator (1/1)
  - Sub Mix (0/16)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

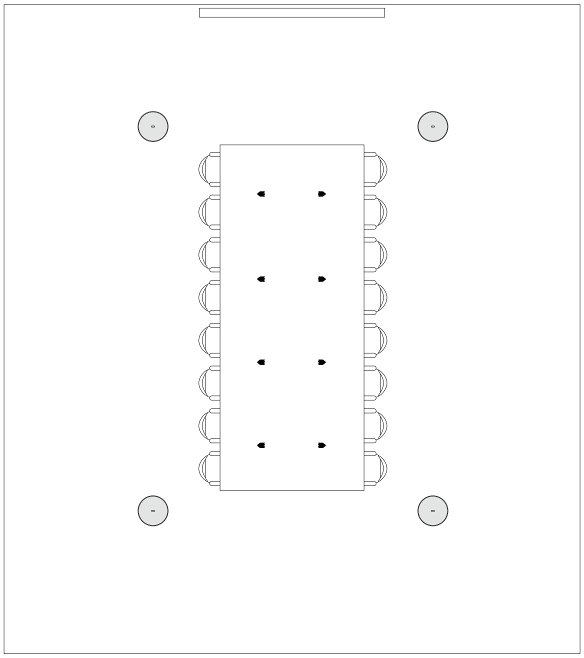
Controlling The System

While a control system can be used to adjust volume levels and to mute the signal paths, this example uses the SoundStation VTX1000 to control the telephone line, muting status of the send signal to the remote telephony participants, and the in room level of the telephone signal.

Creating an Eight Microphones, Video, and Telephony Application Conferencing System

This example creates a typical mono conferencing system with eight table microphones, mono program audio, a mono video codec, and a single audio amplifier zone. The room may look similar to the reflected ceiling plan shown in the following figure with in-ceiling loudspeakers, a video screen in the front of the room, and microphones distributed on the table.

Room Layout for Conferencing System



The block diagram of this system is shown in the following figure. The channel names are labeled with the virtual channel names that are created by default by the SoundStructure Studio software.

Block Diagram of Conferencing System

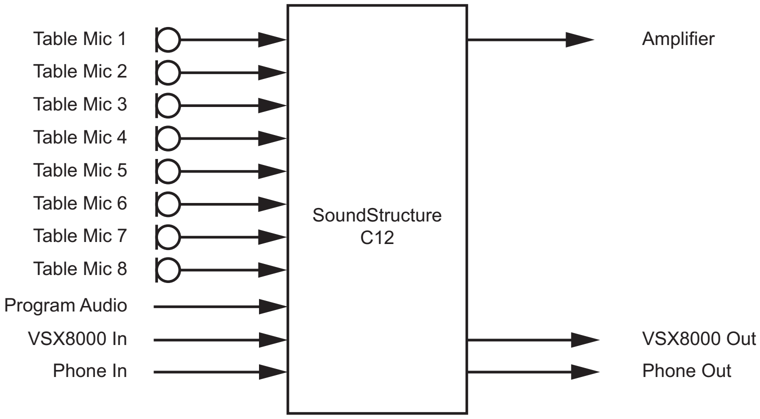


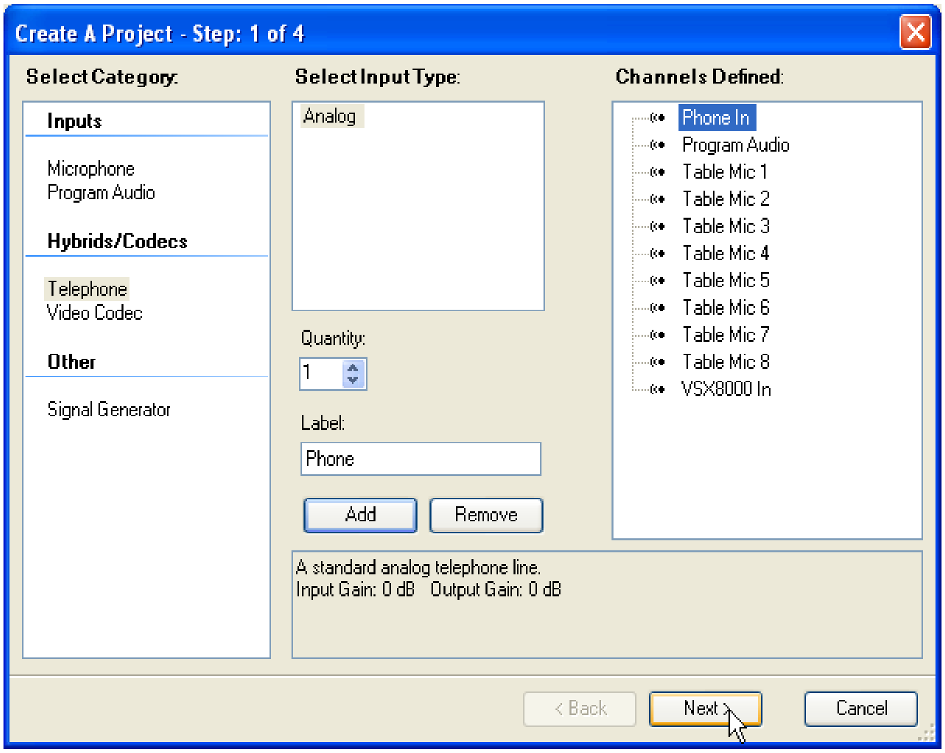
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
Program Audio
VSX8000 In
Phone In
SoundStructure C12
Amplifier
VSX8000 Out
Phone Out

Creating a Project in SoundStructure Studio

The steps to create this project are shown in the following figures. The names for the channels are the names that SoundStructure Studio defines.

Step 1 - Select Inputs

Select eight table microphones, a mono program audio source, a VSX8000 mono video codec, and a telephone interface.


Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Telephone
Video Codec
Other
Signal Generator
Select Input Type:
Analog
Channels Defined:
Phone In
Program Audio
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
VSX8000 In
Add
Remove
< Back
Next >
Cancel

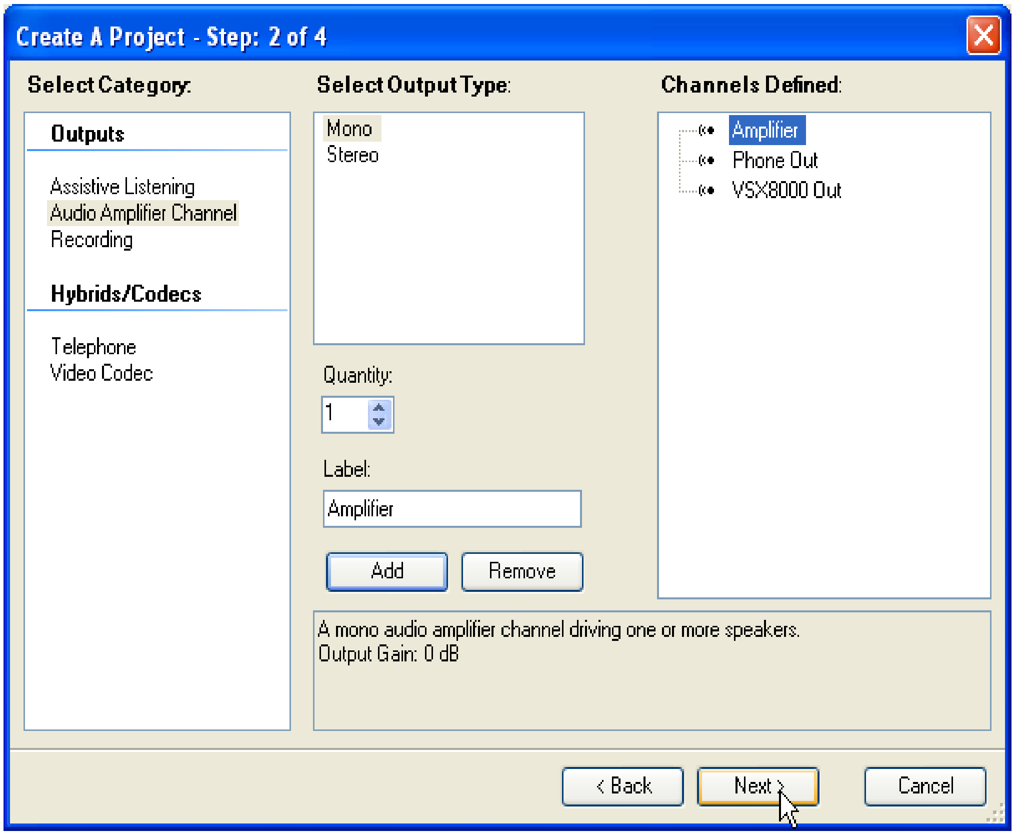
Quantity: 1

Label: Phone

A standard analog telephone line.
Input Gain: 0 dB Output Gain: 0 dB

Step 2 - Select Outputs

Select a mono amplifier as the output source. The telephone and VSX8000 outputs are automatically defined when their respective inputs are selected.



Create A Project - Step: 2 of 4
Select Category:
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Telephone
Video Codec
Select Output Type:
Mono
Stereo
Channels Defined:
Amplifier
Phone Out
VSX8000 Out
Add
Remove
< Back
Next >
Cancel

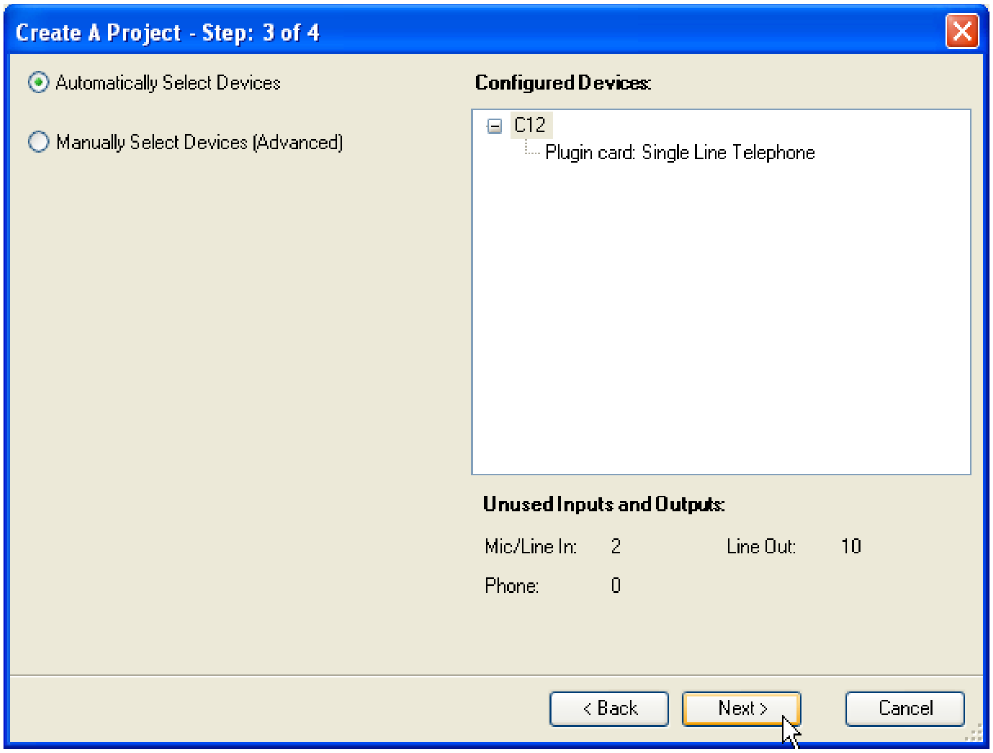
Quantity: 1

Label: Amplifier

A mono audio amplifier channel driving one or more speakers.
Output Gain: 0 dB

Step 3 - Select Devices

Select the equipment required to create this design. By default the SoundStructure C12 with a single line telephone card is selected.



Create A Project - Step: 3 of 4

● Automatically Select Devices

○ Manually Select Devices (Advanced)

Configured Devices:

- C12
  - Plugin card: Single Line Telephone

Unused Inputs and Outputs:

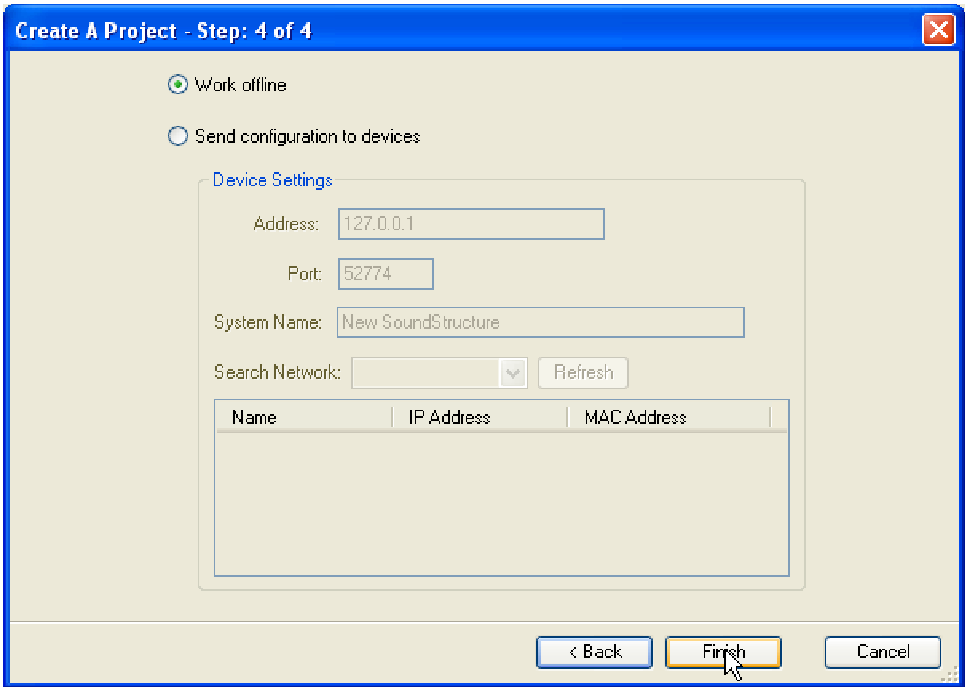
Mic/Line In: 2

Line Out: 10

Phone: 0

Step 4 - Work Offline Or Online

In this step offline operation is selected to create a file for later upload into a SoundStructure C12 and TEL1 single-line telephony card.



Create A Project - Step: 4 of 4

● Work offline

○ Send configuration to devices

Device Settings

Address: 127.0.0.1

Port: 52774

System Name: New SoundStructure

Search Network:

Name	IP Address	MAC Address

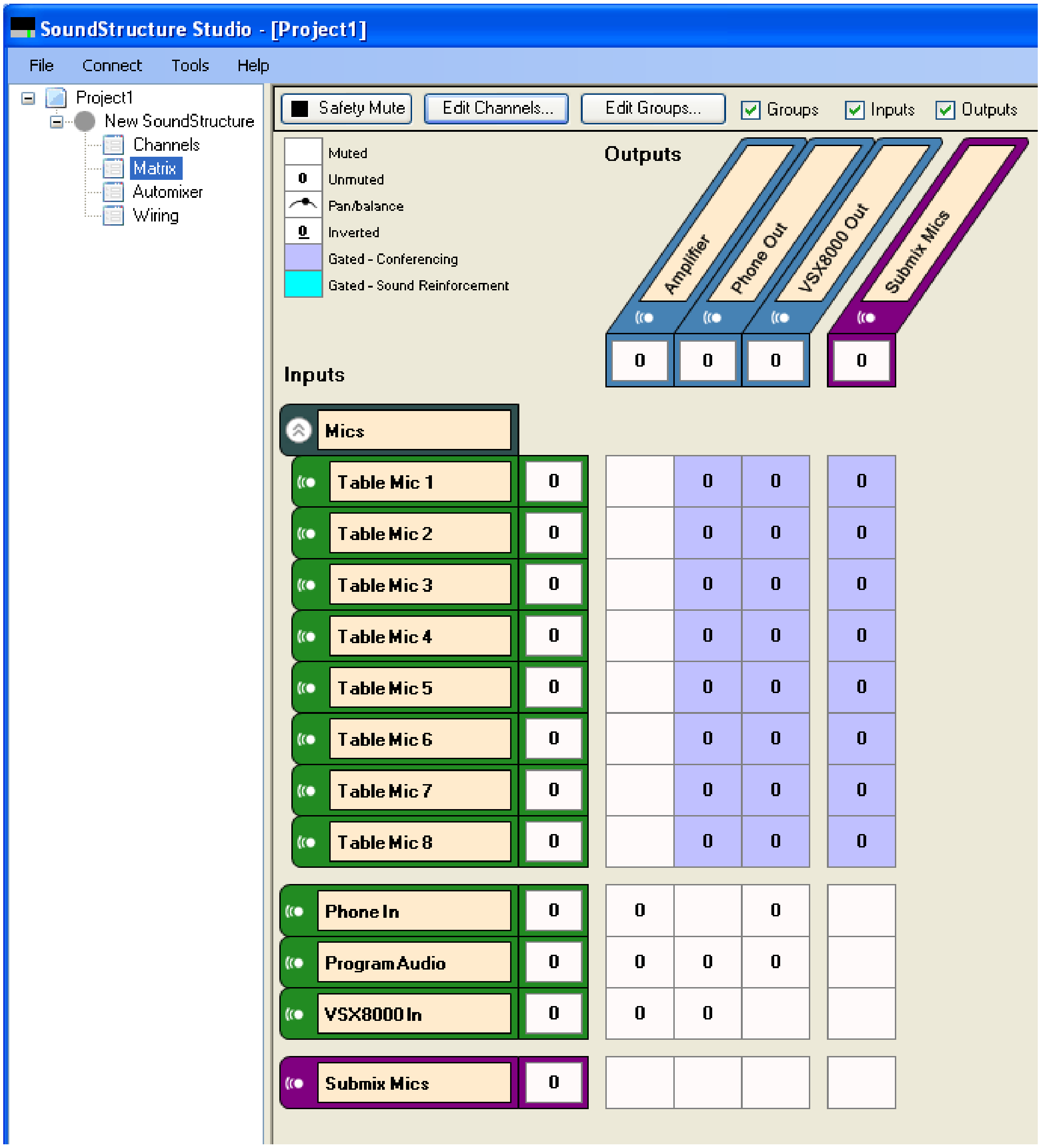
Matrix Settings

Once the system has been designed, click the Matrix label in the project window to view the matrix shown in the following figure.

The input virtual channels that include remote audio are the “Phone In”, “Program Audio”, and “VSX8000 In”. These channels are routed to the “Amplifier” channel so they can be heard in the local room.

The microphones “Table Mic 1” through “Table Mic 8” are routed to the “Phone Out”, “VSX8000 Out”, and “SubMix Mics” channels using the conferencing signal path which includes echo and noise cancellation, and automixer processing. The blue background of these crosspoints is the visual indicator that the conferencing version of the input processing has been selected.

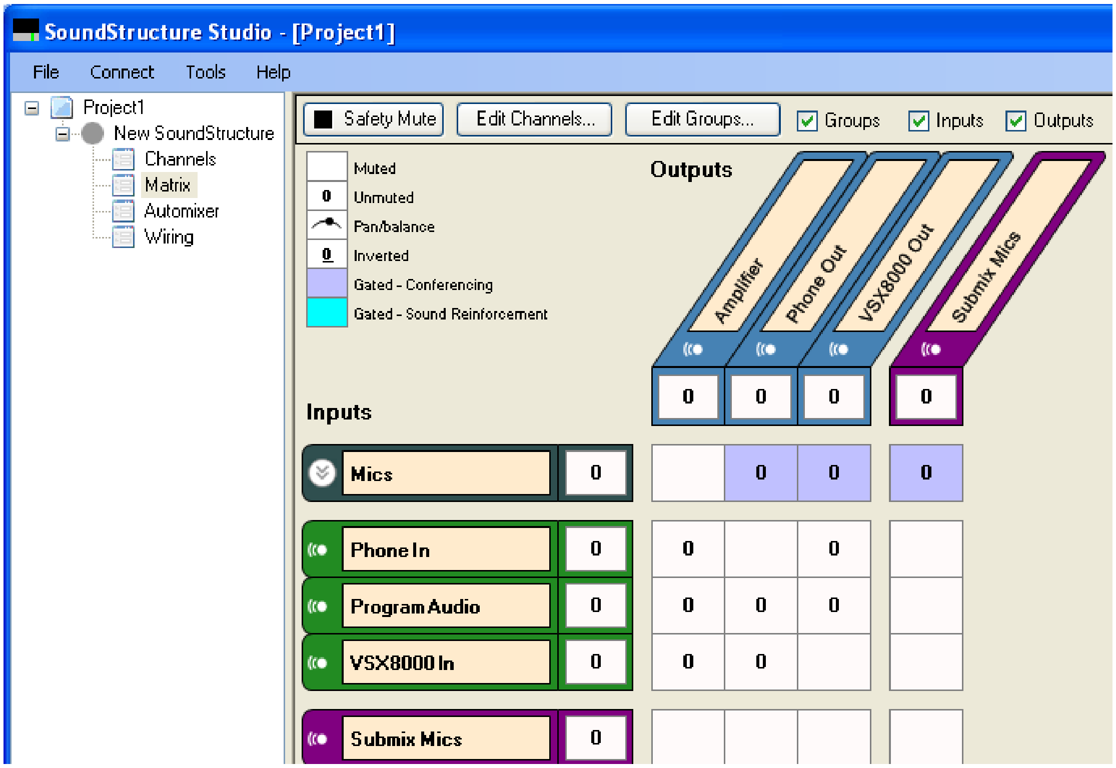
Matrix Label Project Window


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
New SoundStructure
Channels
Matrix
Automixer
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs

Inputs		Amplifier	Phone Out	VSX8000 Out	Submix Mics
		0	0	0	0
Mics					
Table Mic 1	0		0	0	0
Table Mic 2	0		0	0	0
Table Mic 3	0		0	0	0
Table Mic 4	0		0	0	0
Table Mic 5	0		0	0	0
Table Mic 6	0		0	0	0
Table Mic 7	0		0	0	0
Table Mic 8	0		0	0	0
Phone In	0	0		0	
Program Audio	0	0	0	0	
VSX8000 In	0	0	0		
Submix Mics	0				

The matrix may be collapsed by clicking the up arrows next to the “Mics” group. Because all the microphones are used in the same way, the group crosspoint represents how all the table microphone channels are being used. The result is a compact matrix representation as shown in the following figure.

Collapsed View of Matrix Page


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
New SoundStructure
Channels
Matrix
Automixer
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs

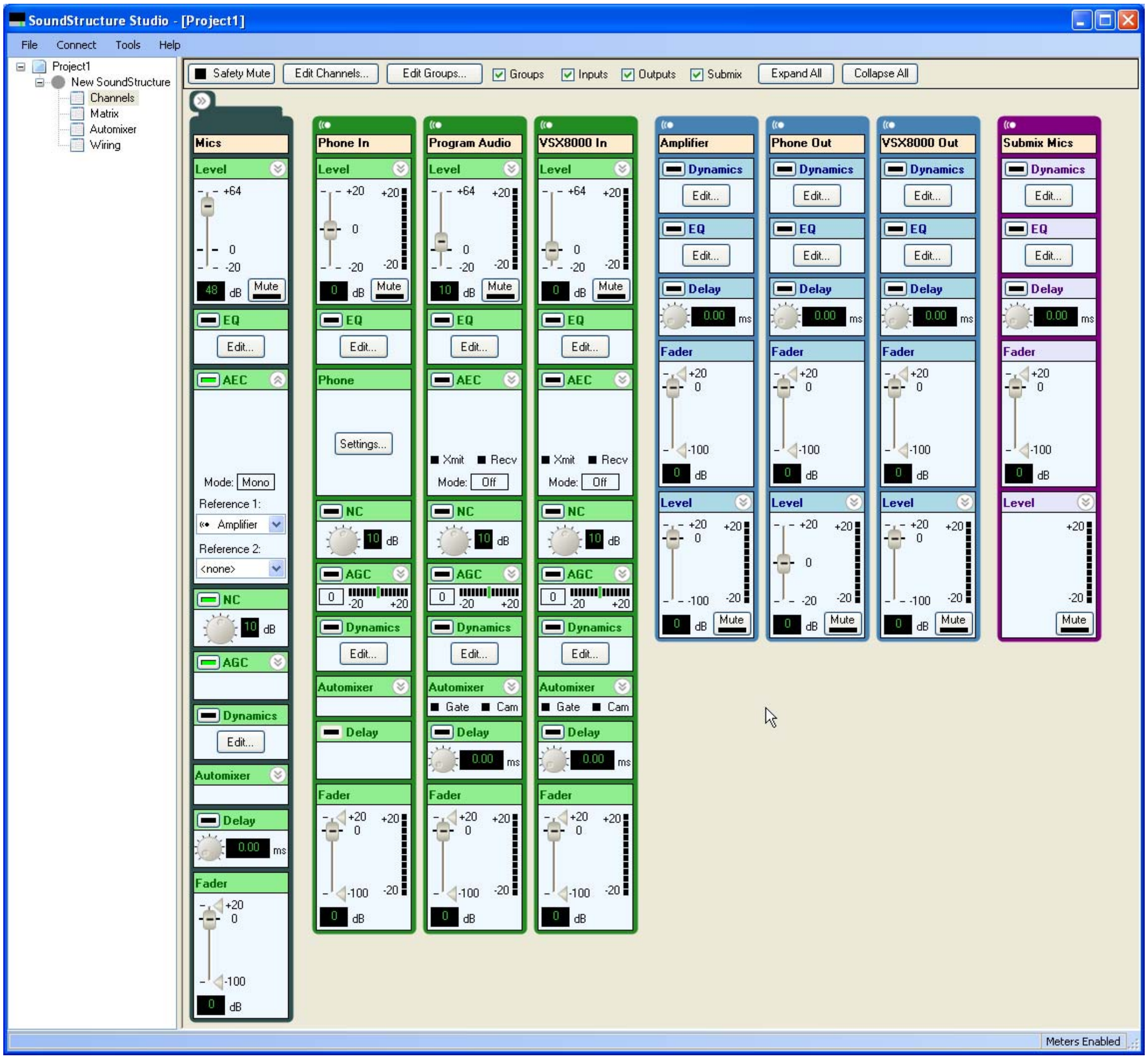
Inputs		Amplifier	Phone Out	VSX8000 Out	Submix Mics
		0	0	0	0
Mics	0		0	0	0
Phone In	0	0		0	
Program Audio	0	0	0	0	
VSX8000 In	0	0	0		
Submix Mics	0				

Channels Settings

The channels page associated with this matrix is shown in the following figure. If the channels are collapsed in the matrix, they are also collapsed in the channels page. The AEC block has been expanded to show the AEC reference.

By default the AEC reference has been set to the mono virtual channel “Amplifier” because this audio includes all the remote audio that need to be echo canceled.

Channels Page for Matrix

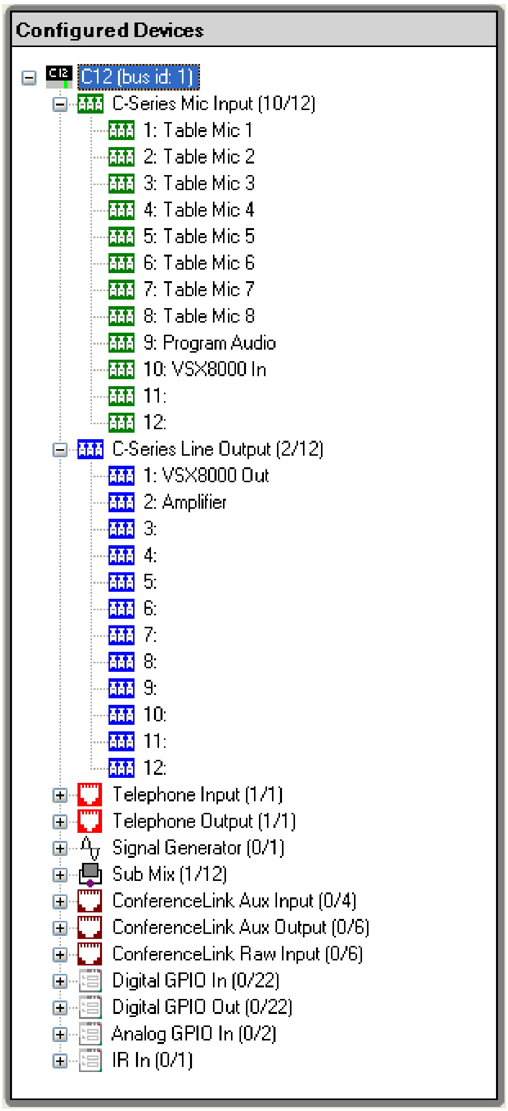

SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
New SoundStructure
Channels
Matrix
Automixer
Wiring
Safety Mute
Edit Channels...
Edit Groups...
Groups
Inputs
Outputs
Submix
Expand All
Collapse All
Mics
Phone In
Program Audio
VSX8000 In
Amplifier
Phone Out
VSX8000 Out
Submix Mics
Level
EQ
AEC
NC
AGC
Dynamics
Automixer
Delay
Fader
Phone
Settings...
Xmit
Recv
Gate
Cam
Edit...
Mute
dB
ms
Meters Enabled

Mode: Mono
Reference 1: Amplifier
Reference 2: <none>

Wiring Information

The system should be wired according to the layout on the wiring page as shown in the following figure. To wire the system with virtual channels on other physical inputs or outputs, simply drag the channels to their desired locations and then wire the system according to the updated wiring information.

Wiring Information



Configured Devices

- C12 (bus id: 1)
  - C-Series Mic Input (10/12)
    - 1: Table Mic 1
    - 2: Table Mic 2
    - 3: Table Mic 3
    - 4: Table Mic 4
    - 5: Table Mic 5
    - 6: Table Mic 6
    - 7: Table Mic 7
    - 8: Table Mic 8
    - 9: Program Audio
    - 10: VSX8000 In
    - 11:
    - 12:
  - C-Series Line Output (2/12)
    - 1: VSX8000 Out
    - 2: Amplifier
    - 3:
    - 4:
    - 5:
    - 6:
    - 7:
    - 8:
    - 9:
    - 10:
    - 11:
    - 12:
  - Telephone Input (1/1)
  - Telephone Output (1/1)
  - Signal Generator (0/1)
  - Sub Mix (1/12)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

Controlling The System

A control system will typically be used to mute microphones and volume settings. The following sections describe how this may be done with the command syntax of the SoundStructure devices. See Appendix A - Command Protocol Reference Guide for additional information on the command set.

Mute

The microphones in the system may be muted either individually or as the “Mics” group by sending the following API command to the SoundStructure device:

```
set mute “Mics” 1
```

will mute all the microphones in the system and

```
set mute “Mics” 0
```

will unmute the microphones in the system.

Volume Control

Volume control in the room can be accomplished by adjusting the fader control on the “Amplifier” virtual channel as follows:

```
inc fader “Amplifier” 1
```

will increase the gain on the “Amplifier” channel by 1dB and

```
dec fader “Amplifier” 1
```

Alternatively the fader settings may be set to an absolute value with the set command as follows:

```
set fader “Amplifier” 0
```

to set the value of the fader to 0dB.

The volume control range can be limited by setting a fader max and fader min as shown in the API syntax below:

```
set fader max “Amplifier” 10
set fader min “Amplifier” -10
```

to limit the maximum and minimum user range of the fader control to +10 and -10dB respectively. The max and min ranges only need to be set once and can be configured as part of the SoundStructure Studio configuration file. If the current amplifier fader setting is outside of this range, the range of the maximum or minimum fader values will be adjusted to include the current fader setting.

Telephone Functions

The telephone interface may be taken offhook by sending the command

```
set phone_connect “Phone Out” 1
```

and placed on hook with the command

```
set phone_connect “Phone Out” 0
```

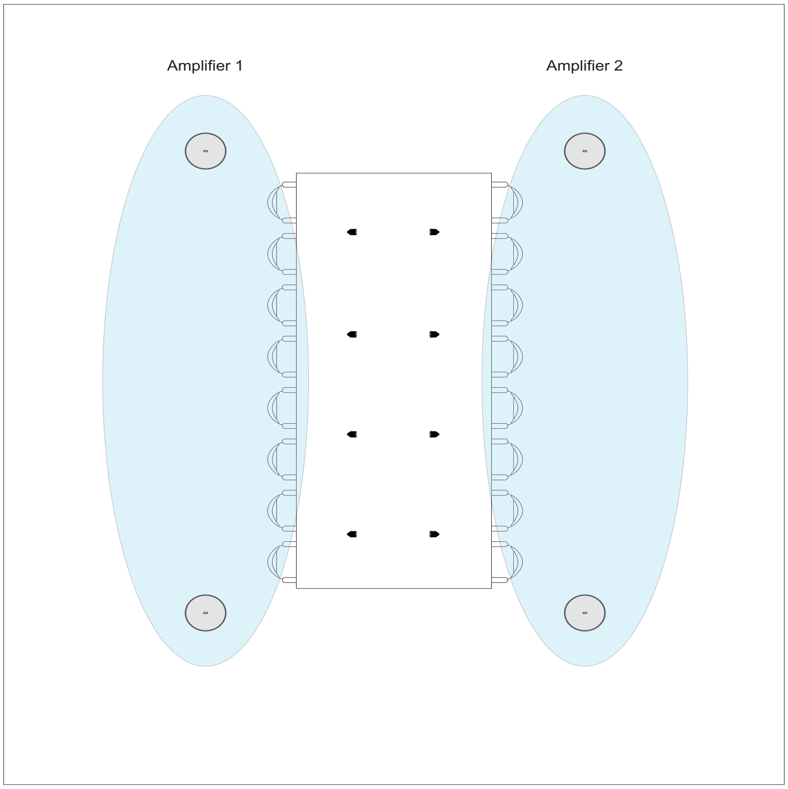
The telephone may be set to dial the digits 1234567, once taken offhook, with the command:

```
set phone_dial “Phone Out” “1234567”
```

Creating a Two PSTN Line Positional “Receive” Audio Conferencing System

This example creates a positional receive audio conferencing system using two telephony lines to represent two remote participants. The system is called “positional receive” because the two remote participants will come from different loudspeakers to create a positional experience where one remote talker comes from one loudspeaker and the other remote talker’s audio is associated with the other loudspeaker system. The layout of the room may look like the following figure with two zones of audio driving the ceiling loudspeakers.

Room Layout for Conferencing System



This system will be designed to include eight table microphones, stereo program audio, two telephony interfaces, and either a stereo amplifier or two mono channel audio amplifiers.

The block diagram of this system is shown in the next figure. The channel names are labeled with the virtual channel names that are created by default by the SoundStructure Studio software.

Block Diagram for Conferencing Room

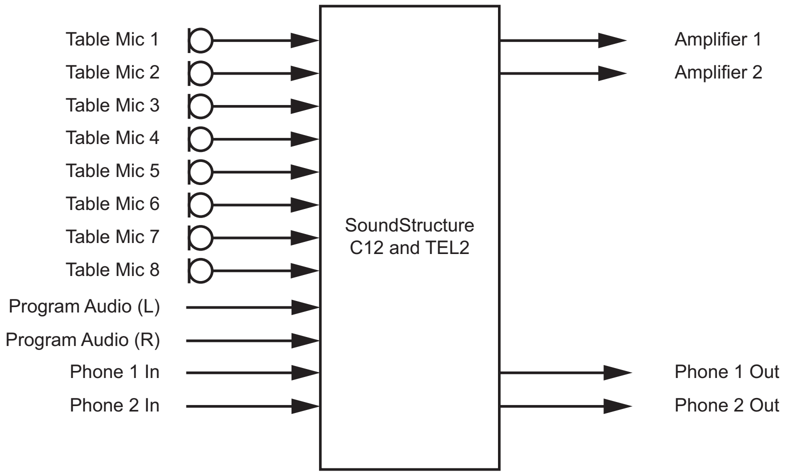


Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
Program Audio (L)
Program Audio (R)
Phone 1 In
Phone 2 In
SoundStructure C12 and TEL2
Amplifier 1
Amplifier 2
Phone 1 Out
Phone 2 Out

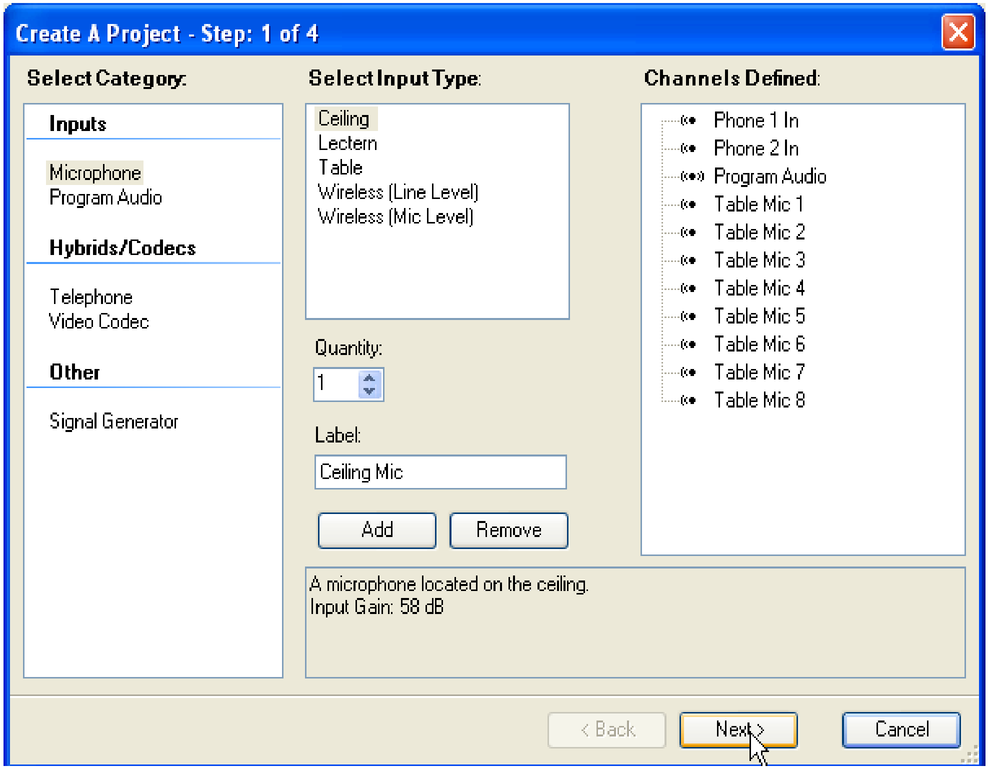
To create this design, start the SoundStructure Studio software and follow the steps shown in the next section.

SoundStructure Studio Steps

The steps to create this project are shown in the following figures. The names for the channels are the default names created by SoundStructure Studio, although the virtual channel names could be set to any valid text string.

Step 1 - Select Inputs

Select 8 table microphones, a stereo program audio source, and two telephone interfaces.


Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Telephone
Video Codec
Other
Signal Generator
Select Input Type:
Ceiling
Lectern
Table
Wireless (Line Level)
Wireless (Mic Level)
Channels Defined:
Phone 1 In
Phone 2 In
Program Audio
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
Add
Remove
< Back
Next >
Cancel

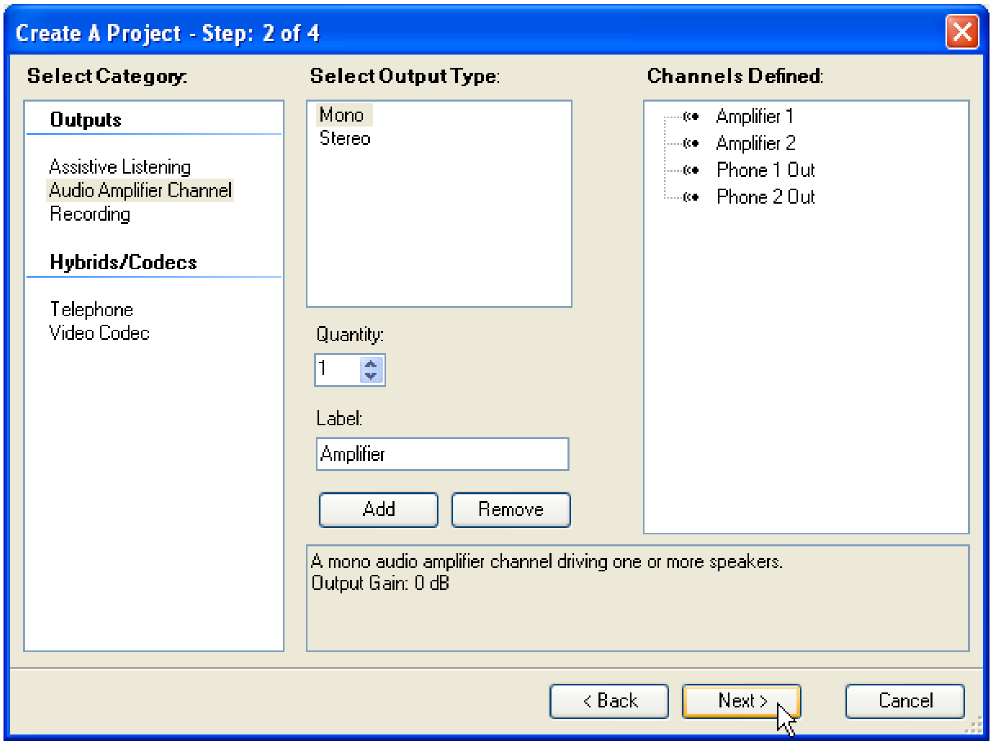
Quantity: 1

Label: Ceiling Mic

A microphone located on the ceiling.
Input Gain: 58 dB

Step 2 - Select Outputs

Select two mono amplifiers as the output devices for this example. The telephone outputs are automatically defined when their respective inputs are added.

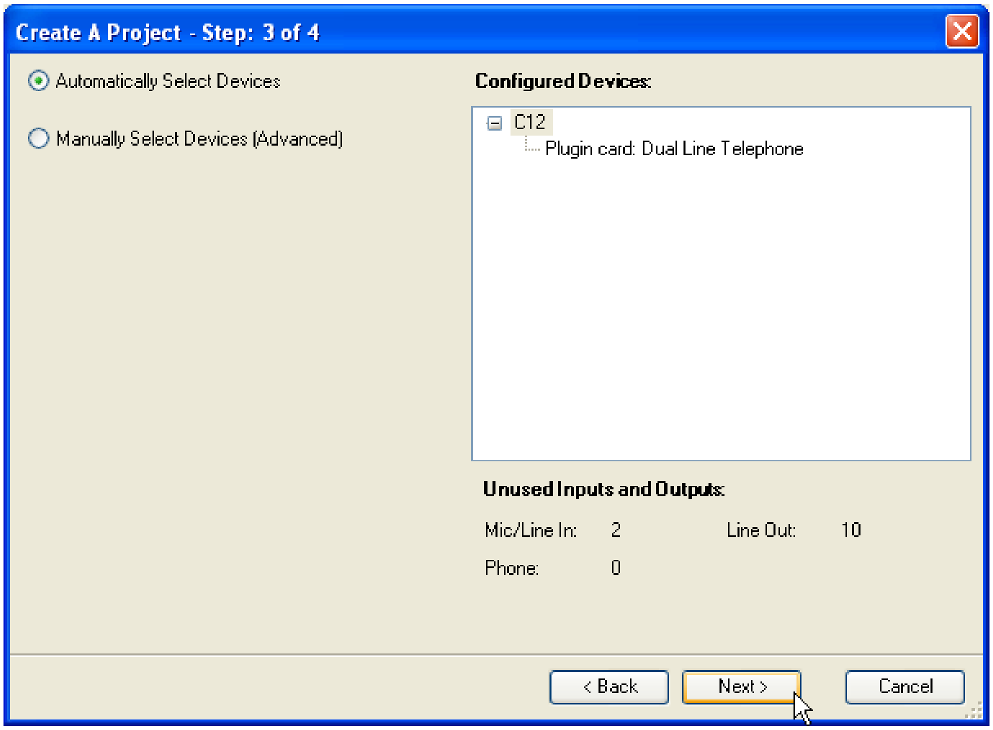


Create A Project - Step: 2 of 4
Select Category:
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Telephone
Video Codec
Select Output Type:
Mono
Stereo
Quantity:
1
Label:
Amplifier
Add
Remove
Channels Defined:
Amplifier 1
Amplifier 2
Phone 1 Out
Phone 2 Out
< Back
Next >
Cancel

A mono audio amplifier channel driving one or more speakers.
Output Gain: 0 dB

Step 3 - Select Equipment

Select the equipment required to create this design. By default the SoundStructure C12 with a dual-line telephone card is selected.


Create A Project - Step: 3 of 4
< Back
Next >
Cancel

● Automatically Select Devices

○ Manually Select Devices (Advanced)

Configured Devices:

⊟ C12
  Plugin card: Dual Line Telephone

Unused Inputs and Outputs:

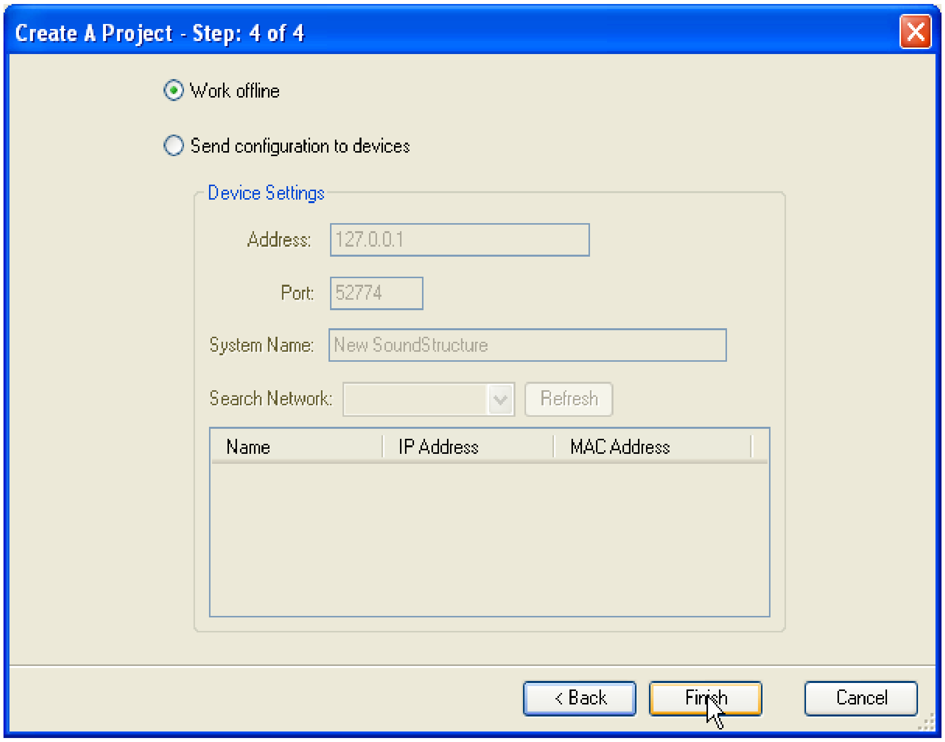
Mic/Line In: 2

Line Out: 10

Phone: 0

Step 4 - Work Offline or Online

In this step offline operation is selected to create a file for later upload into a SoundStructure C12 and dual-line telephony card.



Create A Project - Step: 4 of 4
Device Settings
Name
IP Address
MAC Address
Refresh
< Back
Finish
Cancel

● Work offline

○ Send configuration to devices

Address: 127.0.0.1

Port: 52774

System Name: New SoundStructure

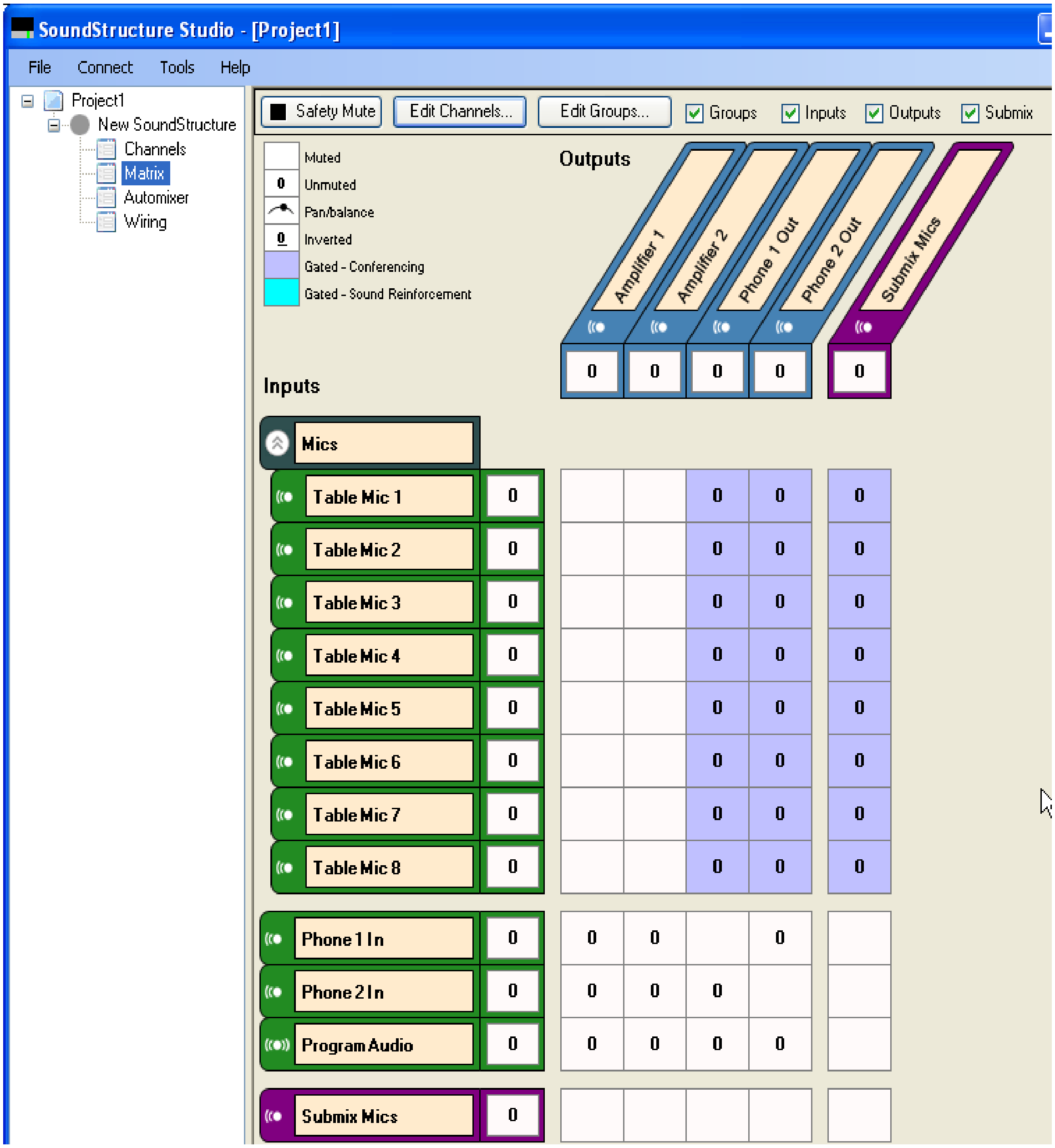
Search Network:

Matrix Settings

Once the system has been designed, select the Matrix entry on the project window on the left tab to view the matrix shown in the following figure.

By default the two telephone lines are routed to both “Amplifier 1” and “Amplifier 2” and the stereo program audio “Program Audio” channel is routed as a mono signal to both Amplifier 1 and Amplifier 2 as shown in the next figure.

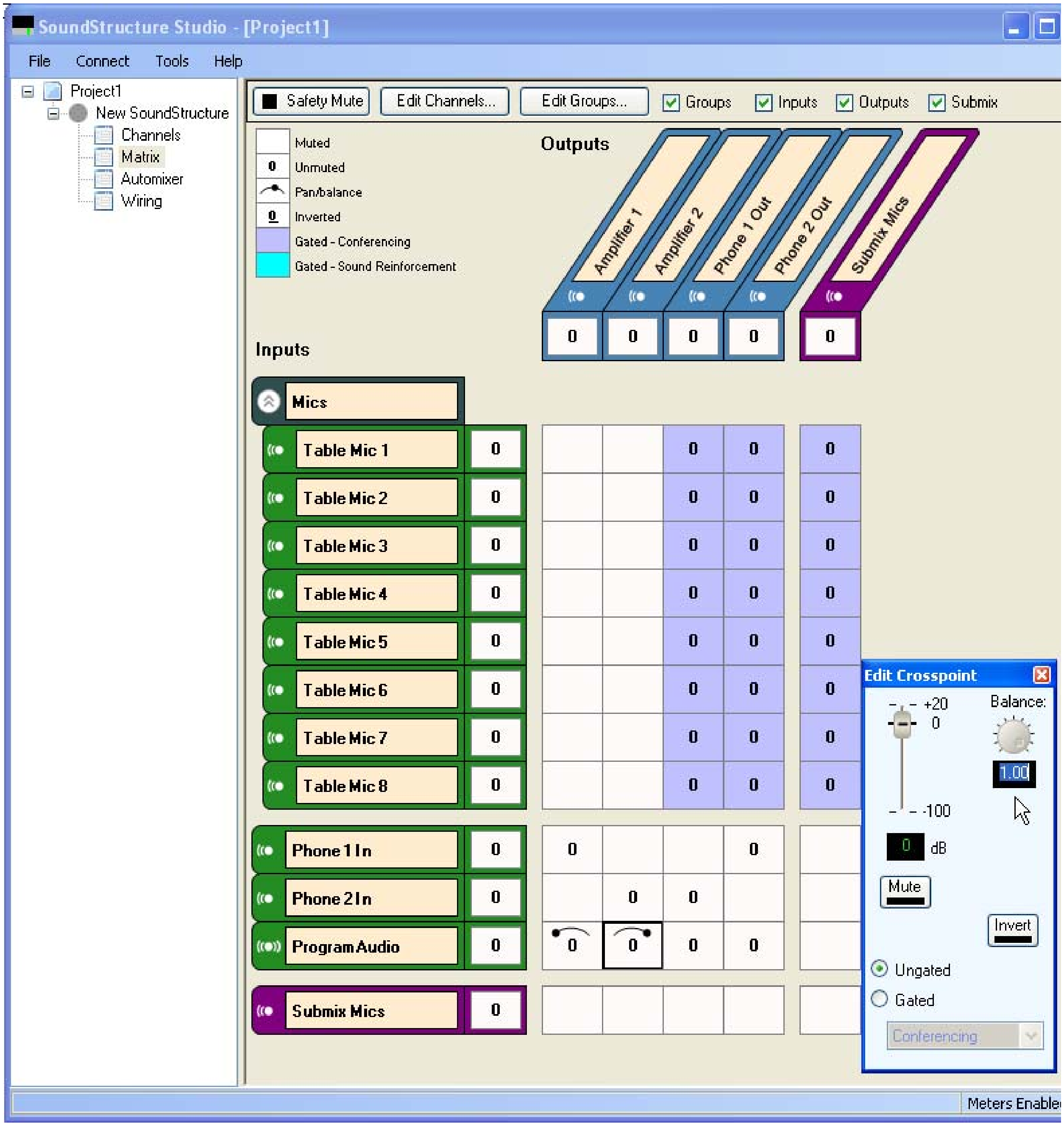
Matrix Project Window.


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
New SoundStructure
Channels
Matrix
Automixer
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Muted
Unmuted
Pan/balance
Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs
Mics

Inputs		Amplifier 1	Amplifier 2	Phone 1 Out	Phone 2 Out	Submix Mics
		0	0	0	0	0
Table Mic 1	0			0	0	0
Table Mic 2	0			0	0	0
Table Mic 3	0			0	0	0
Table Mic 4	0			0	0	0
Table Mic 5	0			0	0	0
Table Mic 6	0			0	0	0
Table Mic 7	0			0	0	0
Table Mic 8	0			0	0	0
Phone 1 In	0	0	0		0	
Phone 2 In	0	0	0	0		
Program Audio	0	0	0	0	0	
Submix Mics	0					

To create the positional solution, route one telephony interface to one amplifier and route the other to the second amplifier. Also, we’ll make the assumption that each amplifier should receive one channel of the stereo program audio. The mapping of the stereo program audio signal to the mono amplifier outputs can be adjusted with the balance control as shown in the following figure. The program audio is balanced to the left to “Amplifier 1” and to the right to “Amplifier 2”.

Mapping of Stereo Program Audio Signal


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
New SoundStructure
Channels
Matrix
Automixer
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Muted
Unmuted
Pan/balance
Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs
Mics
Edit Crosspoint
+20
0
-100
dB
Mute
Invert
Conferencing
Meters Enable

Inputs		Amplifier 1	Amplifier 2	Phone 1 Out	Phone 2 Out	Submix Mics
		0	0	0	0	0
Table Mic 1	0			0	0	0
Table Mic 2	0			0	0	0
Table Mic 3	0			0	0	0
Table Mic 4	0			0	0	0
Table Mic 5	0			0	0	0
Table Mic 6	0			0	0	0
Table Mic 7	0			0	0	0
Table Mic 8	0			0	0	0
Phone 1 In	0	0			0	
Phone 2 In	0		0	0		
Program Audio	0	0	0	0	0	
Submix Mics	0					

Edit Crosspoint

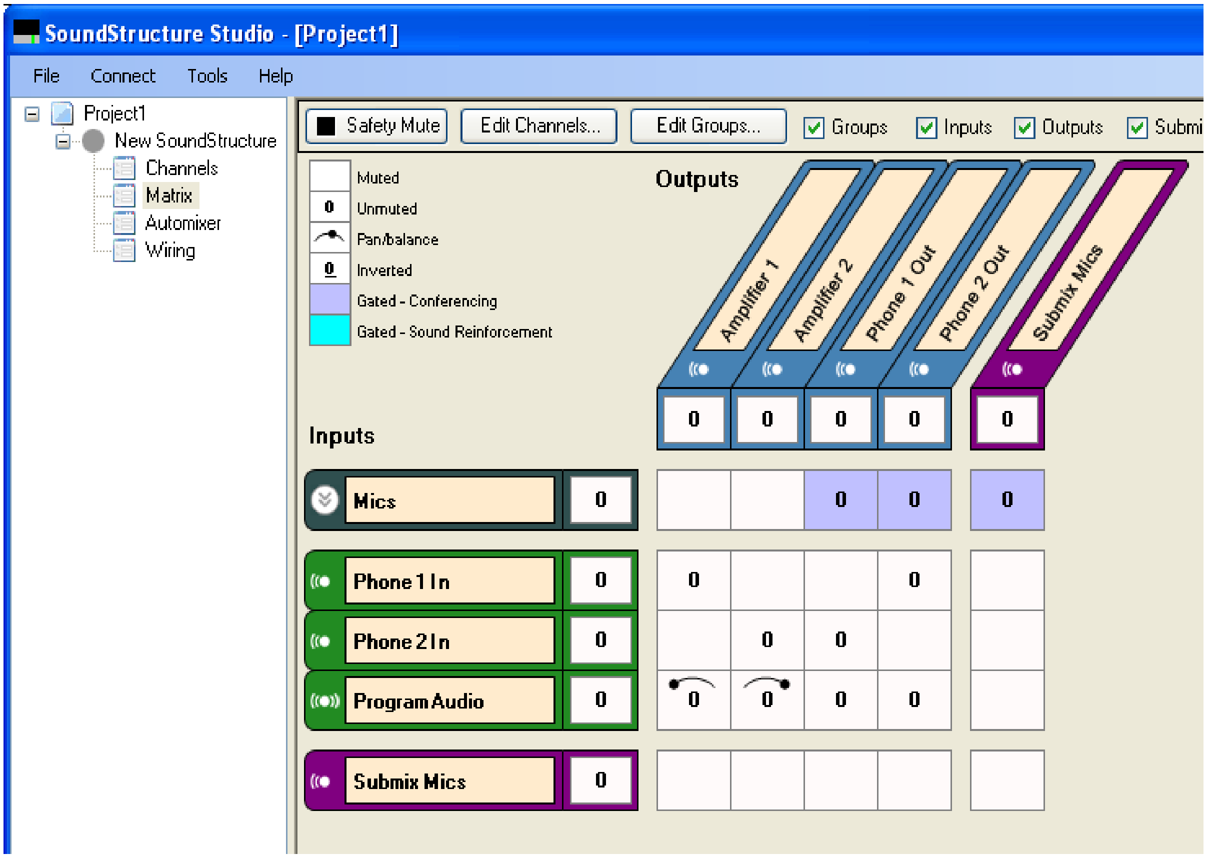
Balance: 1.00

0 dB

● Ungated
○ Gated

The matrix may be collapsed by clicking the arrows next to the “Mics” group resulting in the compact matrix representation shown in the following figure. This figure also shows the routing of each telephony interface to the other telephony interface so that both callers can hear the other caller.

Compact Matrix and Telephony Interface Routing


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
New SoundStructure
Channels
Matrix
Automixer
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submi
Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs

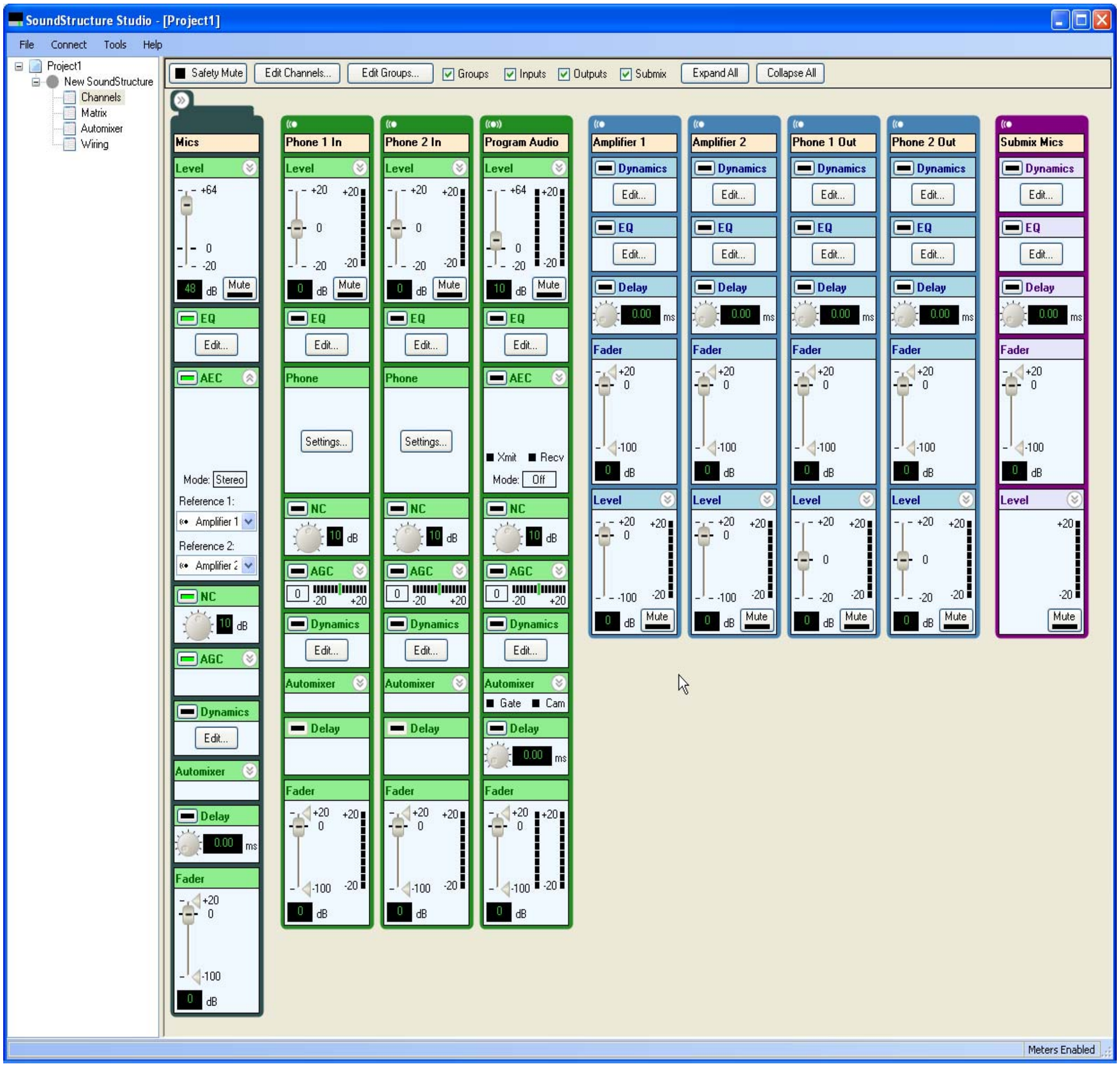
Inputs		Amplifier 1	Amplifier 2	Phone 1 Out	Phone 2 Out	Submix Mics
		0	0	0	0	0
Mics	0			0	0	0
Phone 1 In	0	0			0	
Phone 2 In	0		0	0		
Program Audio	0	0	0	0	0	
Submix Mics	0					

Channels Settings

The channels page will look like the following figure. The AEC block has been expanded to show the AEC references.

By default the two AEC references have been set to the two mono amplifiers “Amplifier 1” and “Amplifier 2” and is then shown to be in stereo mode.

Project Channels Page


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
New SoundStructure
Channels
Matrix
Automixer
Wiring
Safety Mute
Edit Channels...
Edit Groups...
Groups
Inputs
Outputs
Submix
Expand All
Collapse All
Mics
Phone 1 In
Phone 2 In
Program Audio
Amplifier 1
Amplifier 2
Phone 1 Out
Phone 2 Out
Submix Mics
Level
Dynamics
EQ
AEC
Phone
Settings...
NC
AGC
Automixer
Delay
Fader
Edit...
Mute
Xmit
Recv
Gate
Cam
Meters Enabled

Mode: Stereo

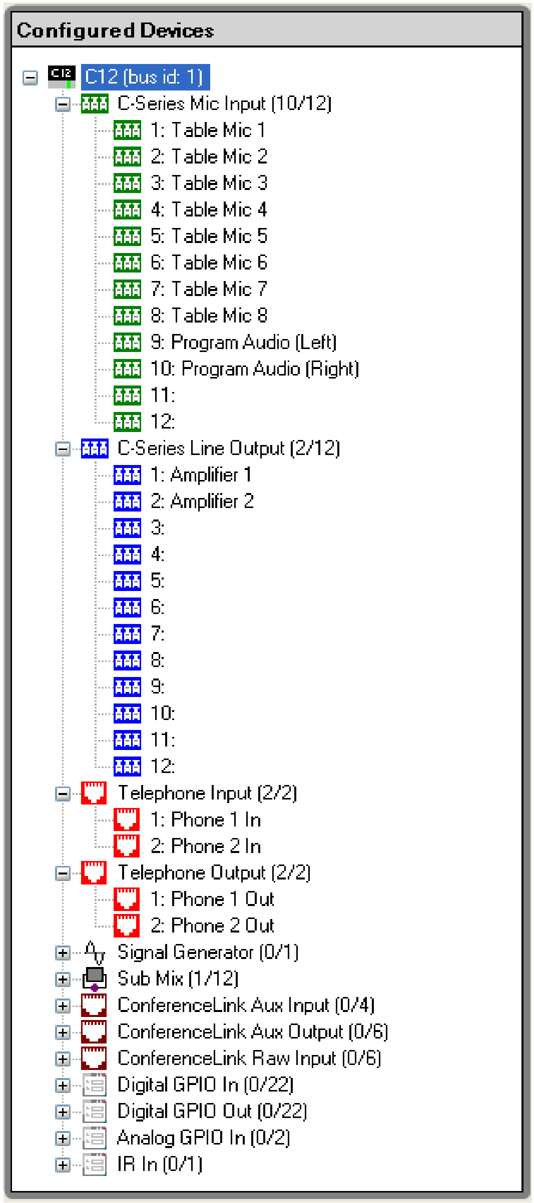
Reference 1: Amplifier 1

Reference 2: Amplifier 2

Wiring Information

The system should be wired according to the information found in the wiring page and shown in the next figure. To wire the system with virtual channels on other physical inputs or outputs, simply drag the channels to their desired locations and then wire the system according to the modified wiring information.

Project Wiring Information



Configured Devices

- C12 (bus id: 1)
  - C-Series Mic Input (10/12)
    - 1: Table Mic 1
    - 2: Table Mic 2
    - 3: Table Mic 3
    - 4: Table Mic 4
    - 5: Table Mic 5
    - 6: Table Mic 6
    - 7: Table Mic 7
    - 8: Table Mic 8
    - 9: Program Audio (Left)
    - 10: Program Audio (Right)
    - 11:
    - 12:
  - C-Series Line Output (2/12)
    - 1: Amplifier 1
    - 2: Amplifier 2
    - 3:
    - 4:
    - 5:
    - 6:
    - 7:
    - 8:
    - 9:
    - 10:
    - 11:
    - 12:
  - Telephone Input (2/2)
    - 1: Phone 1 In
    - 2: Phone 2 In
  - Telephone Output (2/2)
    - 1: Phone 1 Out
    - 2: Phone 2 Out
  - Signal Generator (0/1)
  - Sub Mix (1/12)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

Controlling The System

Mute

The microphones in the system may be muted either individually or as the “Mics” group as follows:

```
set mute “Mics” 1
```

will mute all the microphones in the system and

```
set mute “Mics” 0
```

will unmute the microphones in the system.

Volume Control

Volume control in the room can be accomplished by adjusting the fader control on the “Amplifier 1” and “Amplifier 2” virtual channel as follows:

```
inc fader “Amplifier 1” 1
```

will increase the gain on the “Amplifier 1” channel by 1dB and

```
dec fader “Amplifier 1” 1
```

Alternatively the fader settings may be set to an absolute value with the set command as follows:

```
set “Amplifier 1” fader 0
```

to set the value of the fader to 0dB. Similar commands can be sent to adjust the volume of “Amplifier 2”.

Telephone Functions

The first telephony interface may be taken offhook by sending the command

```
set phone_connect “Phone 1 Out” 1
```

and placed on hook with the command

```
set phone_connect “Phone 1 Out” 0
```

The telephone may be set to dial the digits 1234567, once taken offhook, with the command:

```
set phone_dial “Phone 1 Out” “1234567”
```

Customizing The Phone Routing

If the system has only one telephony caller, the user may wish to have the telephone caller audio come from both sets of loudspeakers. Assuming the first telephony interface is used if there is only one telephone caller, this can be accomplished by unmuting the “Phone 1 In” channel to “Amplifier 2” with the following command.

```
set matrix_mute "Phone 1 In" "Amplifier 2" 0
```

When the second line is answered, the routing can be changed to mute the first phone line to the second amplifier channel as follows.

```
set matrix_mute "Phone 1 In" "Amplifier 2" 1
```

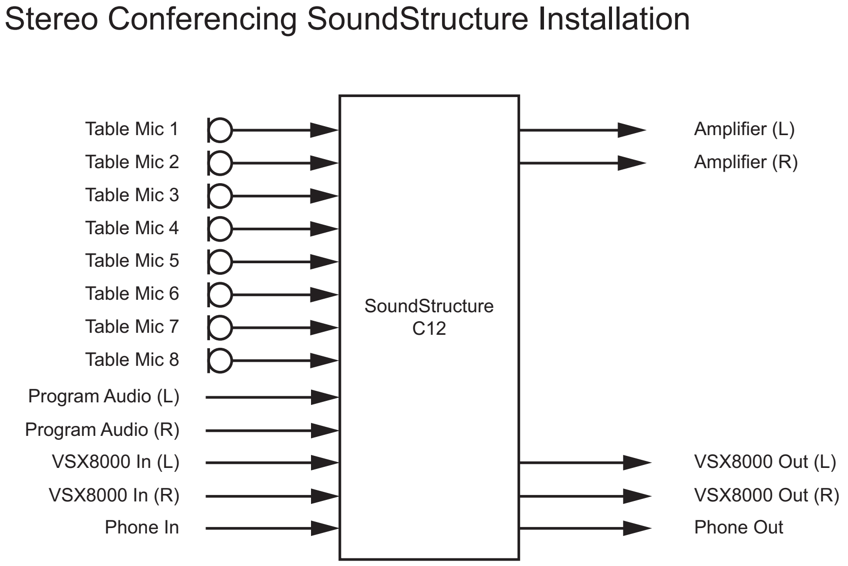
No change to the AEC reference would be required as the AEC reference uses both “Amplifier 1” and “Amplifier 2” and will work whether there is one or two phone lines connected.

Creating an Eight Microphones and Stereo Video Conferencing System

This example creates a stereo video conferencing system with eight table microphones, stereo program audio, a VSX8000 stereo video codec, and a stereo audio amplifier. This application is similar to the 8 microphone mono example shown previously with the addition of the stereo video codec that enables both a positional “receive” signal from the remote site and enables a positional “transmit” signal with the local microphones that can be panned to the two output channels to encode the position of the local talker to the remote participants.

The block diagram of this system is shown in the following figure. The channel names are labeled with the virtual channel names that are created by default by the SoundStructure Studio software.

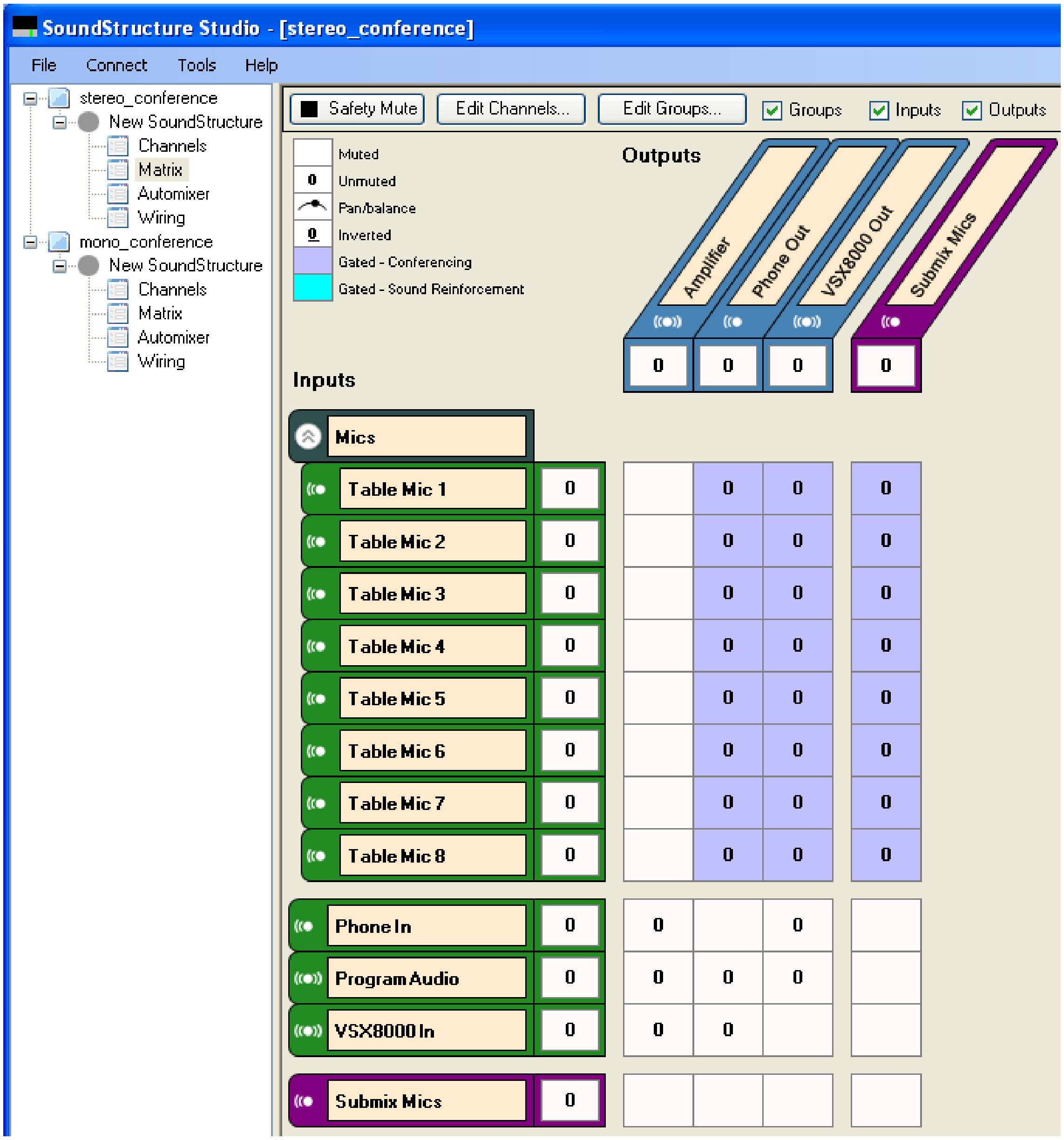
Block Diagram for Conferencing System


Stereo Conferencing SoundStructure Installation
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
Program Audio (L)
Program Audio (R)
VSX8000 In (L)
VSX8000 In (R)
Phone In
SoundStructure C12
Amplifier (L)
Amplifier (R)
VSX8000 Out (L)
VSX8000 Out (R)
Phone Out

The steps to design this configuration are similar to the mono case with the exception of selecting stereo program audio, a stereo VSX8000, and a stereo amplifier.

Once the design is completed, the matrix looks very similar to the mono conferencing case with the exception that the “Program Audio”, “VSX8000 In”, “VSX8000 Out”, and “Amplifier” virtual channels have the stereo graphic symbol next to their names signifying they are stereo virtual channels as shown in the following figure.

Stereo Virtual Channels


SoundStructure Studio - [stereo_conference]
File
Connect
Tools
Help
stereo_conference
New SoundStructure
Channels
Matrix
Automixer
Wiring
mono_conference
New SoundStructure
Channels
Matrix
Automixer
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs

		Amplifier	Phone Out	VSX8000 Out	Submix Mics
		0	0	0	0
Mics					
Table Mic 1	0		0	0	0
Table Mic 2	0		0	0	0
Table Mic 3	0		0	0	0
Table Mic 4	0		0	0	0
Table Mic 5	0		0	0	0
Table Mic 6	0		0	0	0
Table Mic 7	0		0	0	0
Table Mic 8	0		0	0	0
Phone In	0	0		0	
Program Audio	0	0	0	0	
VSX8000 In	0	0	0		
Submix Mics	0				

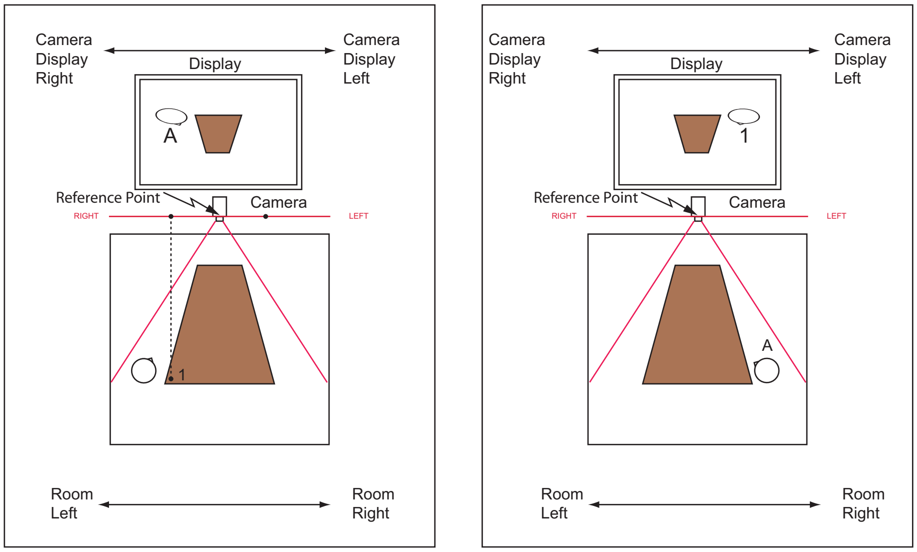
To leverage the stereo capabilities of the VSX8000 codec, it is possible to adjust the panning of the local room microphones to create relative positional information based on the local talker location in the room. This information can be transmitted as part of the stereo audio output signal to the remote participants by adjusting the matrix crosspoint pan settings to reflect the position of the microphones relative to the camera reference point.

Consider the room layout in the following figure that has microphone 1 located at one end of the table at site 1. This microphone has a relative position of being “right” of the camera from the camera’s perspective as shown by the dotted line from the microphone to the camera’s left/right reference line. If you imagine yourself standing where the camera is and looking at the talker at position 1, that talker would be on your right.

The remote participants at site 2 will see the site 1 talker at microphone 1 on the right side of their screen when the remote talkers are looking at the screen because the site 1 talker at microphone 1 is on the “right” side of the camera from the camera’s perspective.

By transmitting positional audio of talker 1 biased to the right channel to the remote site, it is possible to make the local talker at microphone 1 sound as if they were coming from the “room right” loudspeaker to reinforce their visual location as shown in the following figure.

Reinforcing Visual Location of Speaker


Camera Display Right
Display
Camera Display Left
A
Reference Point
Camera
RIGHT
LEFT
1
Room Left
Room Right
Camera Display Right
Display
Camera Display Left
1
Reference Point
Camera
RIGHT
LEFT
A
Room Left
Room Right

Site 1

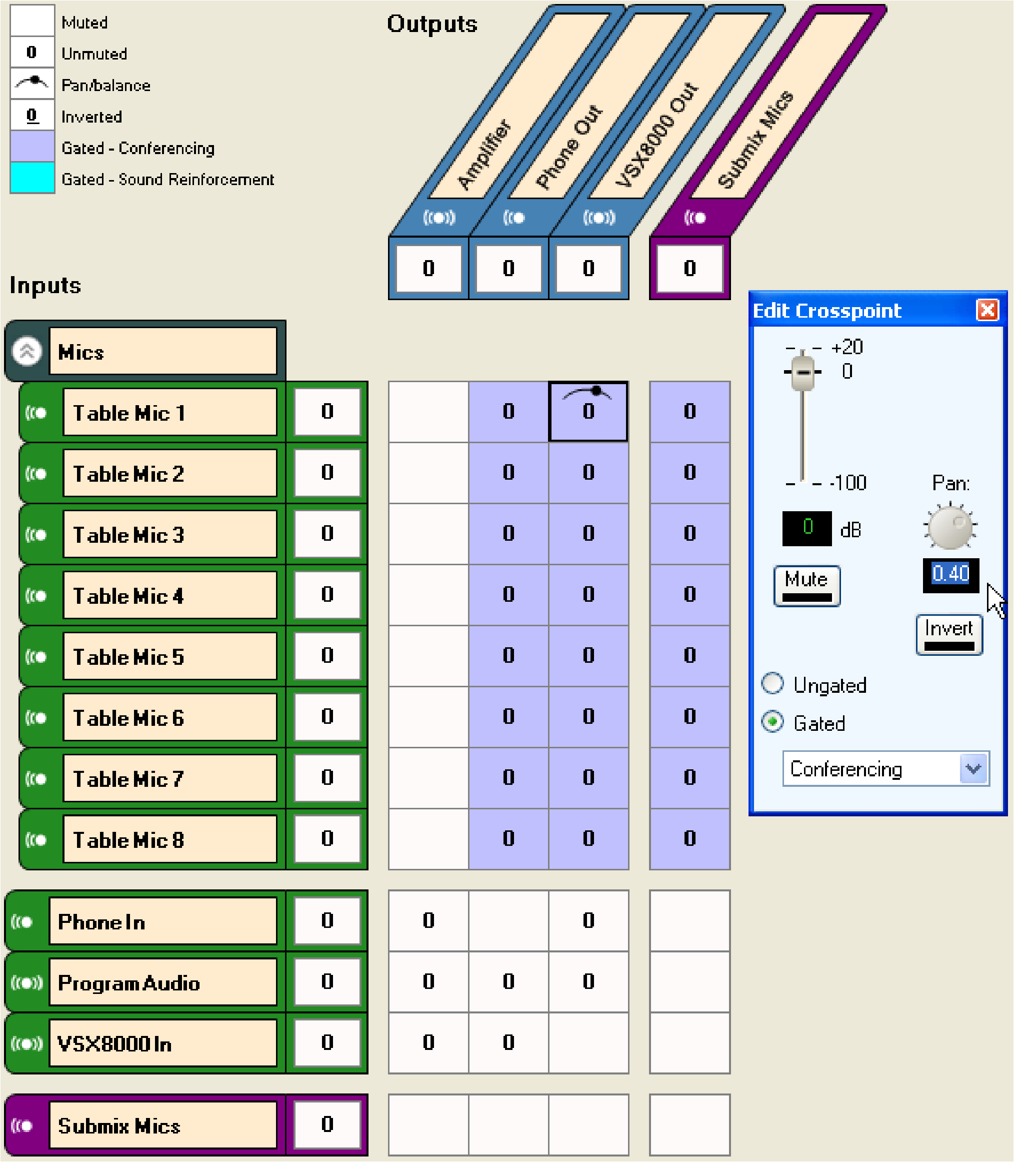
Site 2

The relative position of talker 1 left or right on the screen depends on the relative positioning of the talker with respect to the camera reference point.

To determine the relative positioning relative to the camera, draw a line from the microphone to the camera reference plane as shown in the previous figure. In this example microphone 1 is panned to approximately 0.4 right (assuming the edge of the room is considered 1.0) relative to the camera location. The exact amount of panning can be increased to create a wider spatial presence at the remote site.

The relative position for microphone 1 can be set at the matrix crosspoint to 0.4 as shown in the following figure. This means that the microphone is panned to the right by 0.4.

Setting Relative Position ofr Microphone at Matrix Crosspoint


Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs
Mics

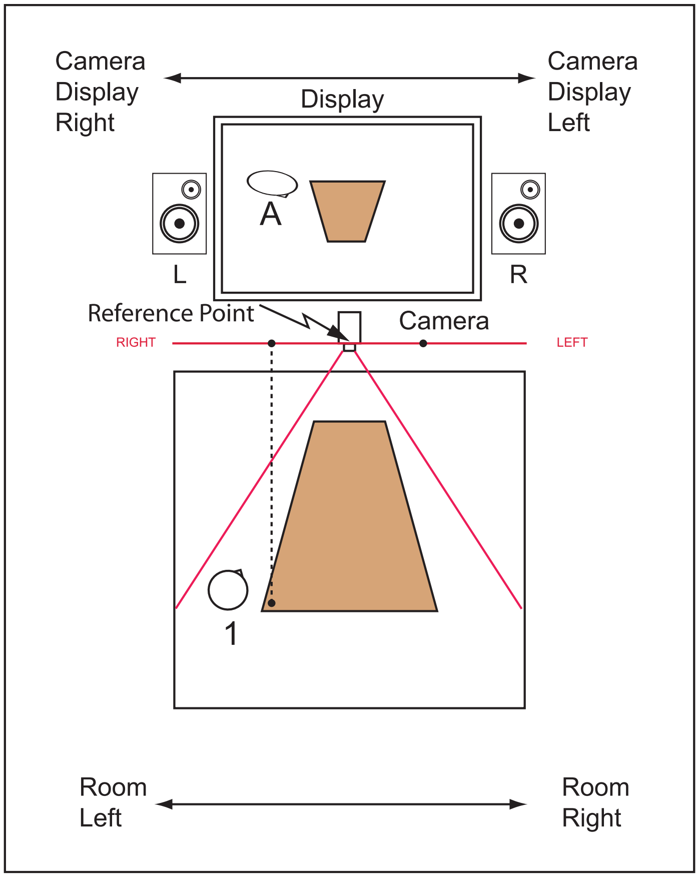
Inputs		Amplifier	Phone Out	VSX8000 Out	Submix Mics
		0	0	0	0
Table Mic 1	0		0	0	0
Table Mic 2	0		0	0	0
Table Mic 3	0		0	0	0
Table Mic 4	0		0	0	0
Table Mic 5	0		0	0	0
Table Mic 6	0		0	0	0
Table Mic 7	0		0	0	0
Table Mic 8	0		0	0	0
Phone In	0	0		0	
Program Audio	0	0	0	0	
VSX8000 In	0	0	0		
Submix Mics	0				

Edit Crosspoint

+20
0
-100
0 dB
Mute
Pan: 0.40
Invert
○ Ungated
● Gated
Conferencing

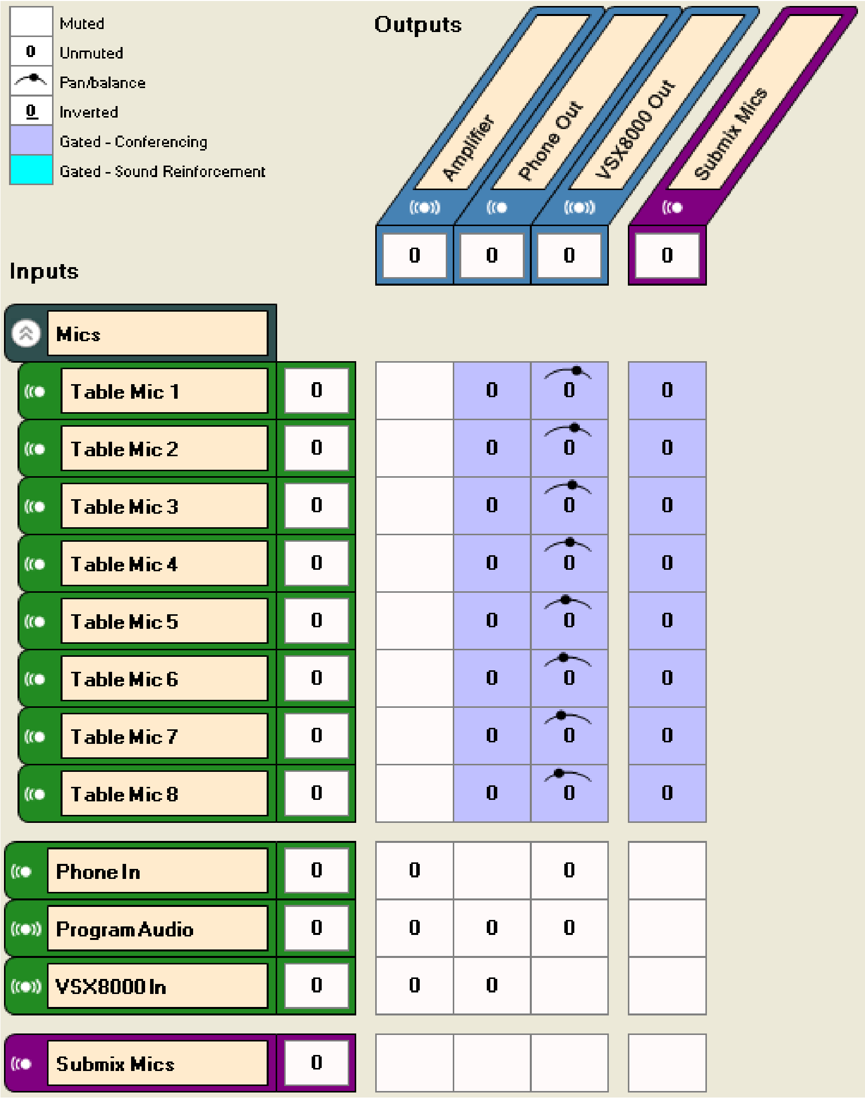
The other microphones also have relative positions as shown in the following figure.

Relative Microphone Positions


Camera Display Right
Camera Display Left
Display
A
L
R
Reference Point
Camera
RIGHT
LEFT
1
Room Left
Room Right

Site 1

By estimating their pan position, the resulting matrix will look like the next figure. As microphones move from right to left relative to the camera, their panning is adjusted from positive to negative.



☐ Muted
0 Unmuted
⌒ Pan/balance
0 Inverted
🟪 Gated - Conferencing
🟦 Gated - Sound Reinforcement

Outputs

Inputs

Inputs		Amplifier	Phone Out	VSX8000 Out	Submix Mics
		0	0	0	0
Mics					
Table Mic 1	0		0	0	0
Table Mic 2	0		0	0	0
Table Mic 3	0		0	0	0
Table Mic 4	0		0	0	0
Table Mic 5	0		0	0	0
Table Mic 6	0		0	0	0
Table Mic 7	0		0	0	0
Table Mic 8	0		0	0	0
Phone In	0	0		0	
Program Audio	0	0	0	0	
VSX8000 In	0	0	0		
Submix Mics	0				

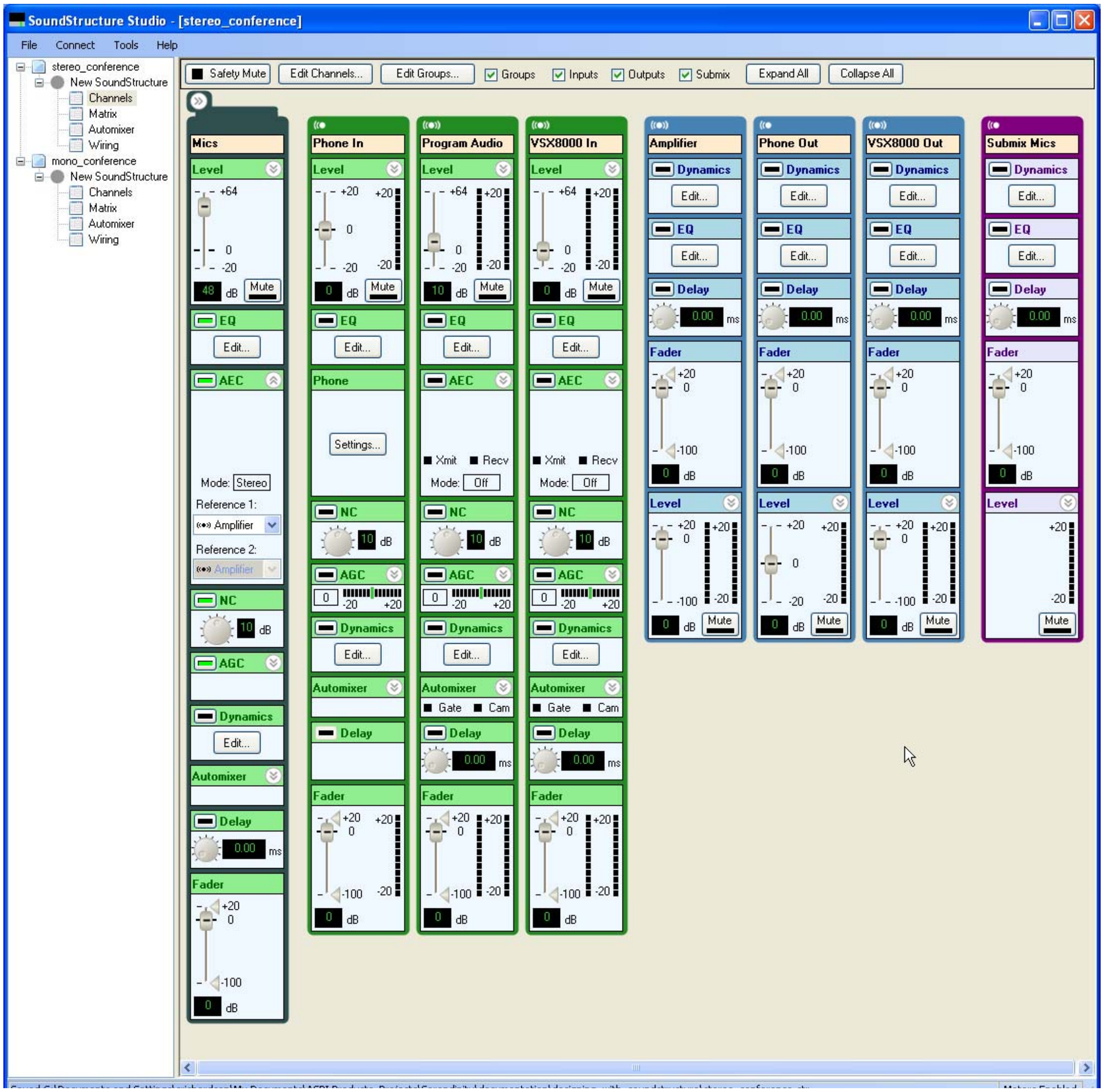
Only the output to the video codec uses the panned output signals because there are two audio channels transmitted to the remote participants. Since the telephony interface is monaural, no panning of the microphones is possible.

Channels Settings

Collapsing the “Mics” group and changing to the channels page will show the screen of the following figure. The AEC block has been expanded to show the AEC reference.

By default the AEC reference has been set to the stereo virtual channel “Amplifier” and is then shown to be in stereo mode.

Collapsed Channel Page


SoundStructure Studio - [stereo_conference]
File
Connect
Tools
Help
stereo_conference
New SoundStructure
Channels
Matrix
Automixer
Wiring
mono_conference
New SoundStructure
Channels
Matrix
Automixer
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Expand All
Collapse All
Mics
Phone In
Program Audio
VSX8000 In
Amplifier
Phone Out
VSX8000 Out
Submix Mics
Level
EQ
AEC
NC
AGC
Dynamics
Automixer
Delay
Fader
Phone
Settings...
Edit...
Mute
Xmit
Recv
Gate
Cam
dB
ms

Mode: Stereo

Reference 1: Amplifier

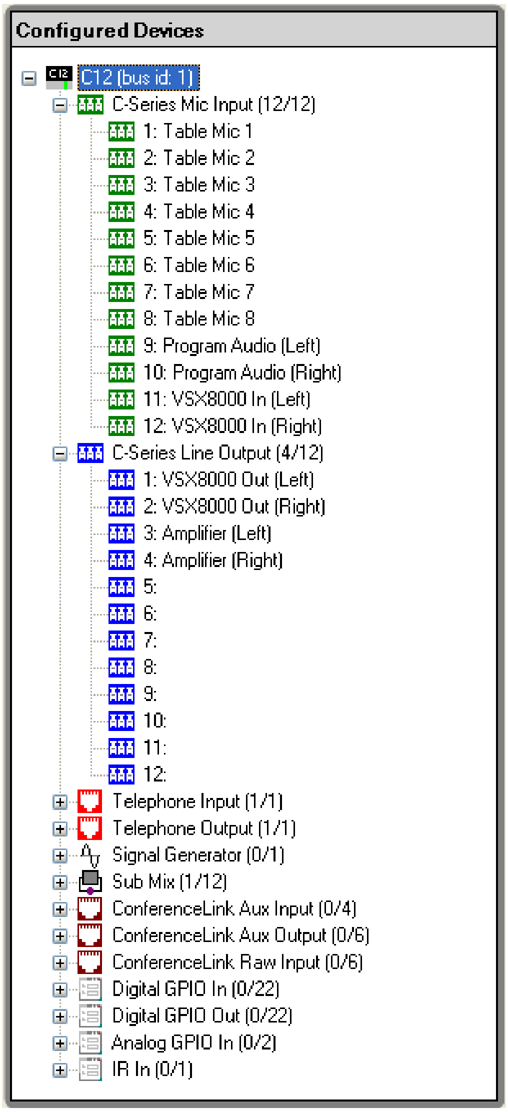
Reference 2: Amplifier

Mode: Off

Wiring Information

The system should be wired according to the information found in the wiring page and shown in the following figure. To wire the system with virtual channels on other physical inputs or outputs, simply drag the channels to their desired locations and then wire the system according to the modified wiring information.

Project Wiring Information



Configured Devices

- C12 (bus id: 1)
  - C-Series Mic Input (12/12)
    - 1: Table Mic 1
    - 2: Table Mic 2
    - 3: Table Mic 3
    - 4: Table Mic 4
    - 5: Table Mic 5
    - 6: Table Mic 6
    - 7: Table Mic 7
    - 8: Table Mic 8
    - 9: Program Audio (Left)
    - 10: Program Audio (Right)
    - 11: VSX8000 In (Left)
    - 12: VSX8000 In (Right)
  - C-Series Line Output (4/12)
    - 1: VSX8000 Out (Left)
    - 2: VSX8000 Out (Right)
    - 3: Amplifier (Left)
    - 4: Amplifier (Right)
    - 5:
    - 6:
    - 7:
    - 8:
    - 9:
    - 10:
    - 11:
    - 12:
  - Telephone Input (1/1)
  - Telephone Output (1/1)
  - Signal Generator (0/1)
  - Sub Mix (1/12)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

Controlling The System

The control of the stereo system is exactly the same as the control of the mono conferencing system. Because the stereo virtual channel names have the same name as the mono virtual channels in the previous example, the SoundStructure API will seamlessly operate on the stereo virtual channel without having to make any change to the control system code.

Creating an Eight Microphones with The Polycom Video Codec Conferencing System

This example shows how to use 8 analog microphones with a SoundStructure device connected to a Polycom HDX video conferencing system. This system will use the telephony interface that is native to the Polycom HDX system.

A drawing of this type of system is shown in the following figure.

Block Diagram for Conferencing System

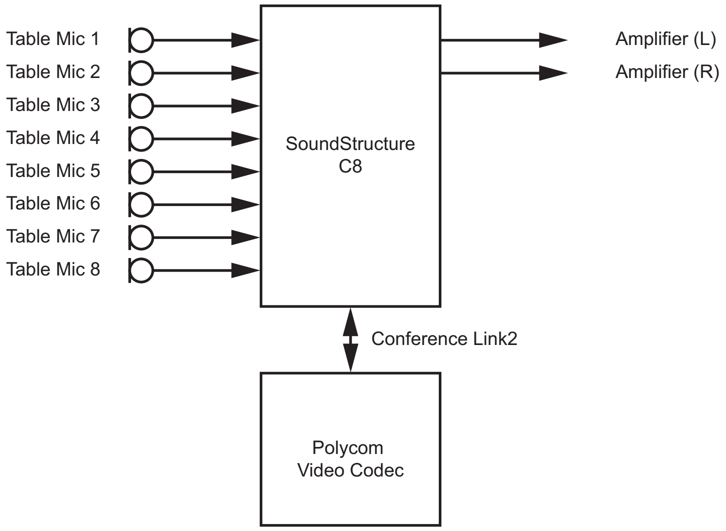


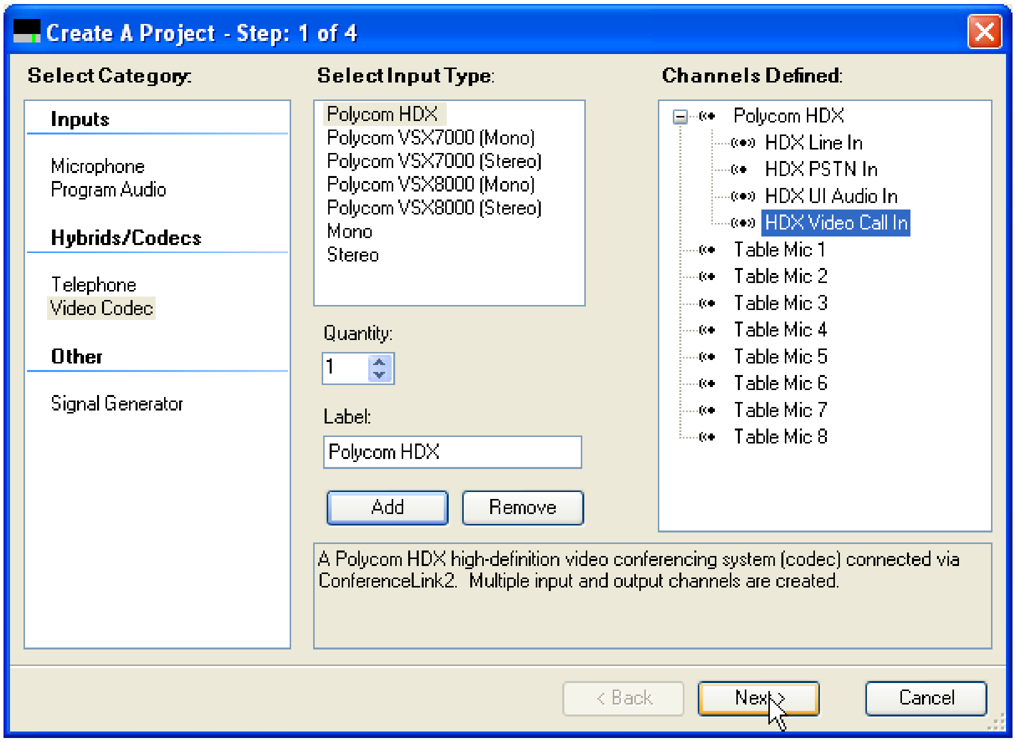
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
SoundStructure C8
Amplifier (L)
Amplifier (R)
Conference Link2
Polycom Video Codec

SoundStructure Studio Steps

The steps to create this project are shown in the next figures. The names for the channels are the names that SoundStructure Studio defines.

Step 1 - Select Inputs

Select eight table microphones and a Polycom HDX video conferencing system. Notice that when the HDX system is selected, there are multiple audio streams that will be transmitted from the HDX to the SoundStructure. Additional information may be found in Connecting Over Conference Link2.



Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Telephone
Video Codec
Other
Signal Generator
Select Input Type:
Polycom HDX
Polycom VSX7000 (Mono)
Polycom VSX7000 (Stereo)
Polycom VSX8000 (Mono)
Polycom VSX8000 (Stereo)
Mono
Stereo
Quantity:
Label:
Add
Remove
Channels Defined:
Polycom HDX
HDX Line In
HDX PSTN In
HDX UI Audio In
HDX Video Call In
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
< Back
Next >
Cancel

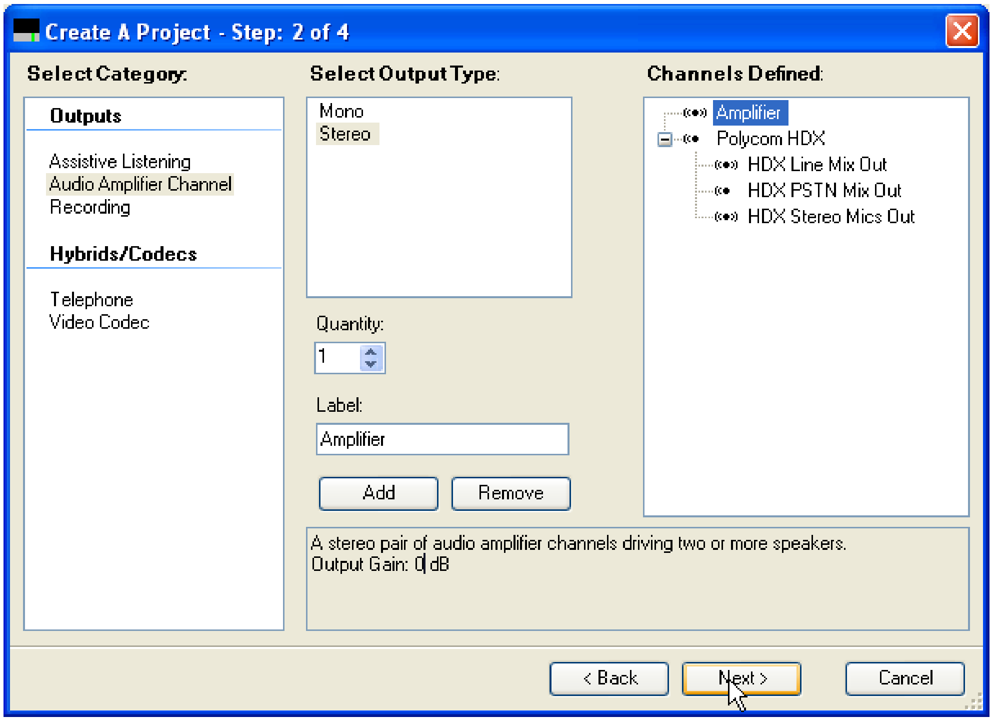
Quantity: 1

Label: Polycom HDX

A Polycom HDX high-definition video conferencing system (codec) connected via ConferenceLink2. Multiple input and output channels are created.

Step 2 - Select Outputs

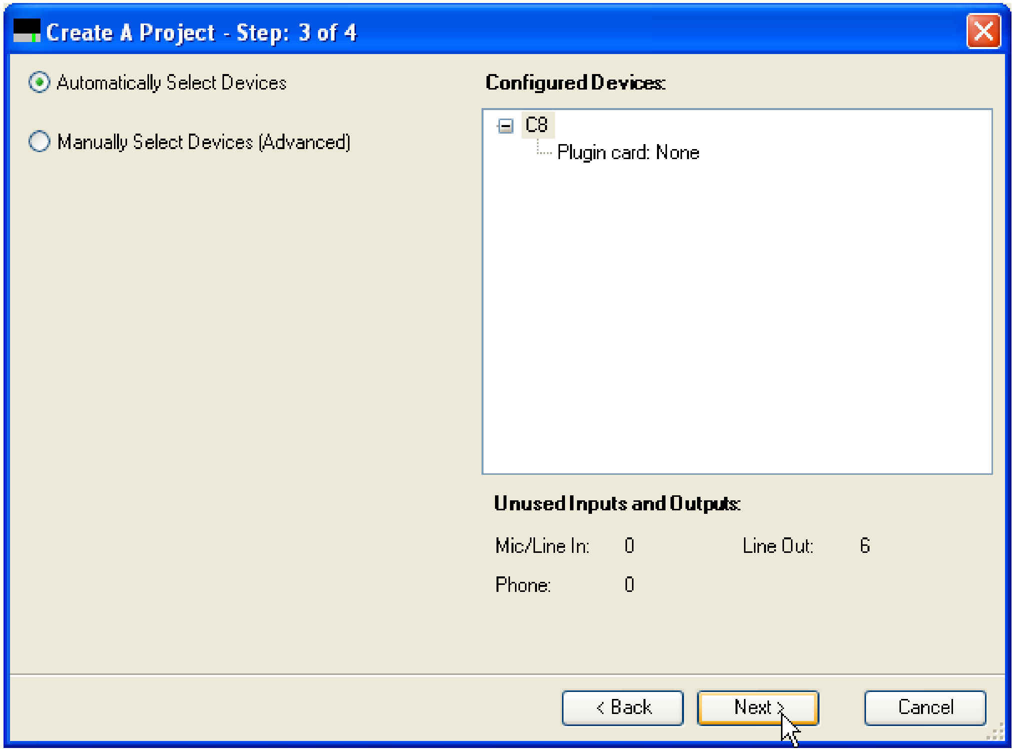
Select a stereo amplifier as the output source. Notice that the Polycom HDX is already defined as an output and includes multiple audio streams that will be sent to the HDX from the SoundStructure device.


Create A Project - Step: 2 of 4
Select Category:
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Telephone
Video Codec
Select Output Type:
Mono
Stereo
Quantity:
1
Label:
Amplifier
Add
Remove
Channels Defined:
Amplifier
Polycom HDX
HDX Line Mix Out
HDX PSTN Mix Out
HDX Stereo Mics Out
< Back
Next >
Cancel

A stereo pair of audio amplifier channels driving two or more speakers.
Output Gain: 0 dB

Step 3 - Select Devices

Select the equipment required to create this design. By default the SoundStructure C8 is selected.



Create A Project - Step: 3 of 4

● Automatically Select Devices

○ Manually Select Devices (Advanced)

Configured Devices:

- C8
  - Plugin card: None

Unused Inputs and Outputs:

Mic/Line In: 0

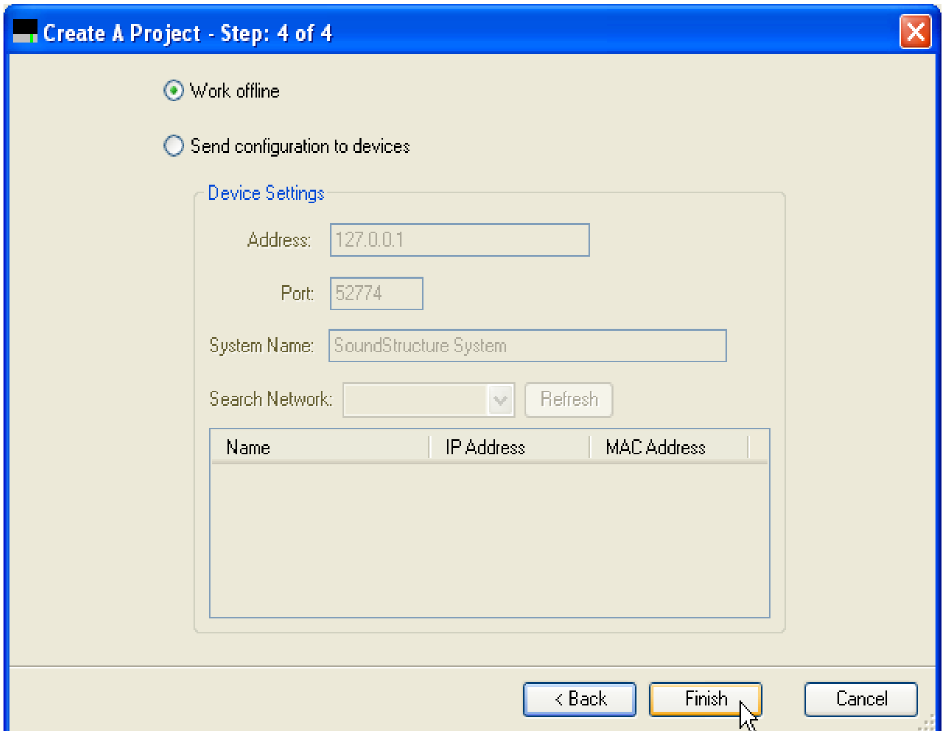
Line Out: 6

Phone: 0

< Back
Next >
Cancel

Step 4 - Work Offline Or Online

In this step offline operation is selected to create a file for later upload into a SoundStructure C8.


Create A Project - Step: 4 of 4
Device Settings
Search Network:
Refresh
Name
IP Address
MAC Address
< Back
Finish
Cancel

● Work offline

○ Send configuration to devices

Address: 127.0.0.1

Port: 52774

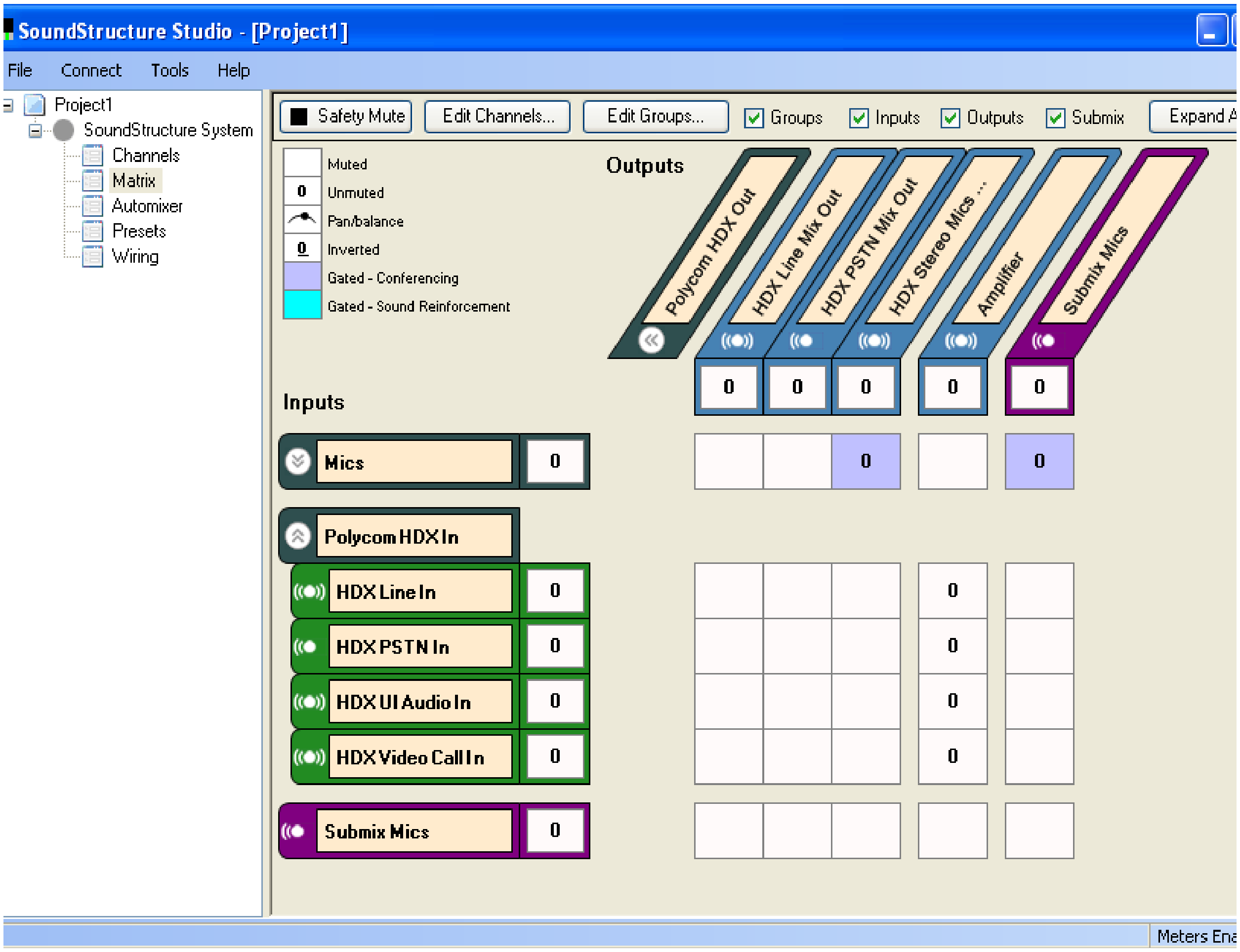
System Name: SoundStructure System

Matrix Settings

Once the system has been designed, click the Matrix label in the project window to view the matrix shown in the following figure.

The input virtual channels include microphones that are included in the virtual channel group “Mics” collapsed as shown in the next figure and the remote audio from the Polycom HDX. The Polycom HDX audio channels are routed to the “Amplifier” channel so they can be heard in the local room, and the echo canceled microphones are routed to the Polycom HDX stereo mics stream so they can be sent to the remote video participants

.Project Matrix Settings


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Expand A
Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs
Meters Ena

	Polycom HDX Out: HDX Line Mix Out	Polycom HDX Out: HDX PSTN Mix Out	Polycom HDX Out: HDX Stereo Mics ...	Amplifier	Submix Mics
	0	0	0	0	0
Mics 0			0		0
Polycom HDX In					
HDX Line In 0				0	
HDX PSTN In 0				0	
HDX UI Audio In 0				0	
HDX Video Call In 0				0	
Submix Mics 0					

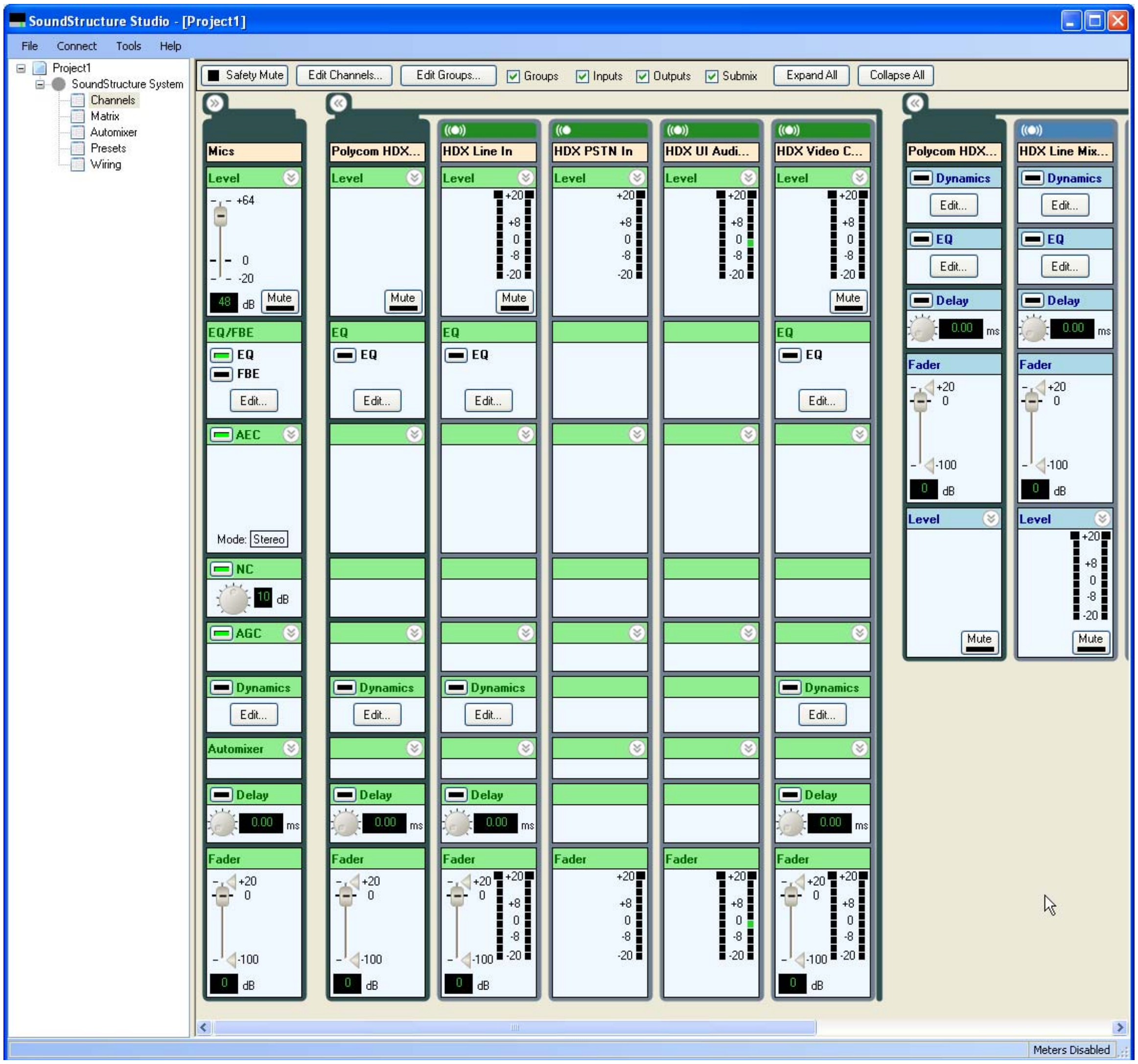
The audio channels from the Polycom HDX system are available as separate audio streams within the SoundStructure matrix.

Channels Settings

The channels page associated with this matrix is shown in the following figure. If the channels are collapsed in the matrix, they are also collapsed in the channels page. The AEC block has been expanded to show the AEC reference.

By default the AEC reference has been set to the mono virtual channel “Amplifier” because this audio includes all the remote audio that need to be echo canceled.

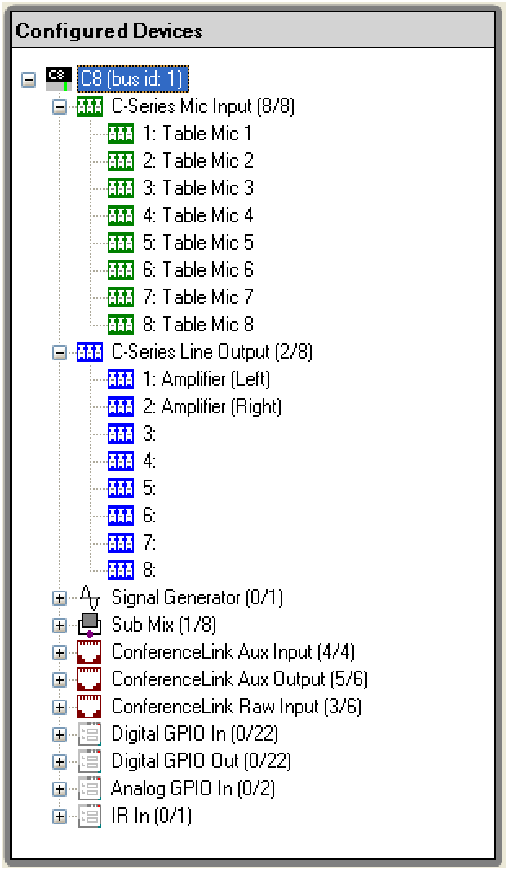
Project Channels Page


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Edit Groups...
Groups
Inputs
Outputs
Submix
Expand All
Collapse All
Mics
Polycom HDX...
HDX Line In
HDX PSTN In
HDX UI Audi...
HDX Video C...
Polycom HDX...
HDX Line Mix...
Level
EQ/FBE
EQ
FBE
AEC
Mode: Stereo
NC
AGC
Dynamics
Automixer
Delay
Fader
Mute
Edit...
Meters Disabled

Wiring Information

The system should be wired according to the layout on the wiring page as shown in the following figure. To wire the system with virtual channels on other physical inputs or outputs, simply drag the channels to their desired locations and then wire the system according to the updated wiring information.

Project Wiring Information



Configured Devices

- C8 (bus id: 1)
  - C-Series Mic Input (8/8)
    - 1: Table Mic 1
    - 2: Table Mic 2
    - 3: Table Mic 3
    - 4: Table Mic 4
    - 5: Table Mic 5
    - 6: Table Mic 6
    - 7: Table Mic 7
    - 8: Table Mic 8
  - C-Series Line Output (2/8)
    - 1: Amplifier (Left)
    - 2: Amplifier (Right)
    - 3:
    - 4:
    - 5:
    - 6:
    - 7:
    - 8:
  - Signal Generator (0/1)
  - Sub Mix (1/8)
  - ConferenceLink Aux Input (4/4)
  - ConferenceLink Aux Output (5/6)
  - ConferenceLink Raw Input (3/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

Controlling The System

A control system will typically be used to mute microphones and volume settings. The following sections describe how this may be done with the command syntax of the SoundStructure devices. See Appendix A - Command Protocol Reference Guide for additional information on the command set.

Mute

The microphones in the system may be muted either individually or as the “Mics” group by sending the following API command to the SoundStructure device:

```
set mute “Mics” 1
```

will mute all the microphones in the system and

```
set mute “Mics” 0
```

will unmute the microphones in the system.

When connected to the Polycom HDX system, the microphones on the SoundStructure by muting the microphones on the Polycom HDX system. As described in Connecting Over Conference Link2, the HDX will send a mute command to the “Mics” group whenever the HDX receives a command to mute via the HDX API or via the HDX IR remote receiver.

Volume Control

Volume control in the room can be accomplished by adjusting the fader control on the “Amplifier” virtual channel as follows:

```
inc fader “Amplifier” 1
```

will increase the gain on the “Amplifier” channel by 1dB and

```
dec fader “Amplifier” 1
```

Alternatively the fader settings may be set to an absolute value with the set command as follows:

```
set fader “Amplifier” 0
```

to set the value of the fader to 0dB.

When connected to the Polycom HDX system, the Amplifier fader setting on the SoundStructure will be adjusted when the volume on the Polycom HDX is adjusted. As described in Connecting Over Conference Link2, the HDX will send a fader command to the “Amplifier” group whenever the HDX receives a command to adjust volume via the HDX API or via the HDX IR remote receiver.

Telephony

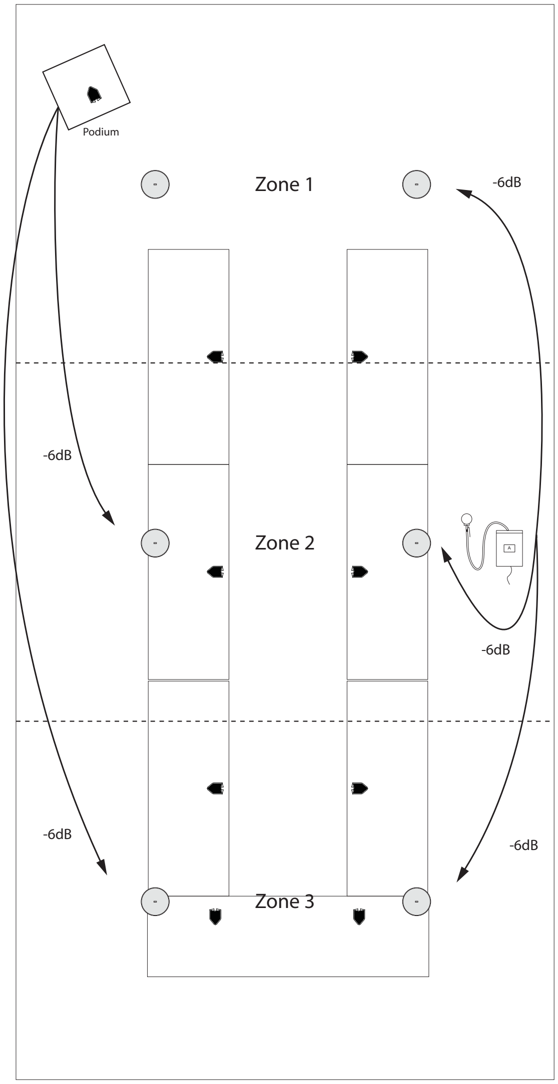
The SoundStructure in this example can use the Polycom HDX’s telephony signal as that is a separate stream that is sent from the HDX to the SoundStructure device. The telephony system would be controlled with the Polycom HDX system.

Creating an Eight Microphones with Wireless and Lectern Microphones Reinforcement Conferencing System

This example shows how to use the sound reinforcement and conferencing processing to create an audio conferencing solution that includes both a lectern and wireless microphone for in-room reinforcement of the presenters’ microphones and use of these microphones for conferencing in addition to tabletop microphones. This example includes eight table microphones, a lectern microphone, a wireless microphone, stereo program audio, a single telephony interface, and three zones of audio amplifiers for reinforcement.

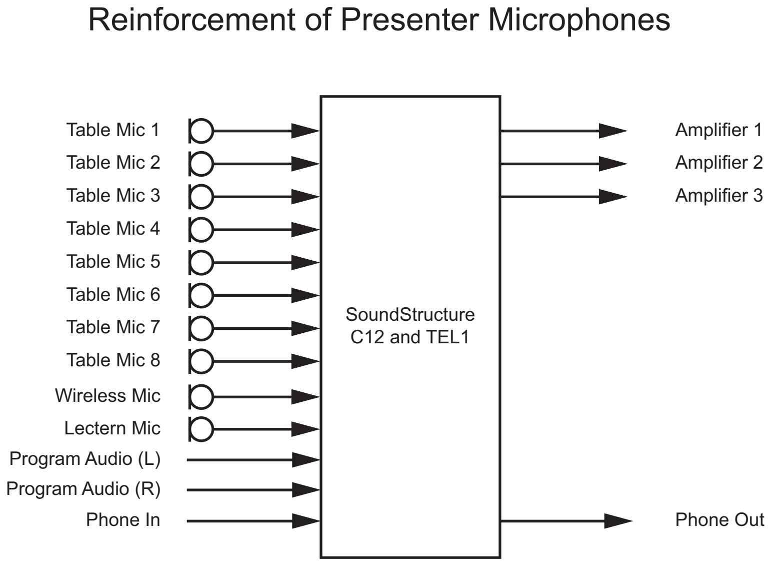
The layout for this style of room can be seen in the following figure. This figure also shows the desired reinforcement levels from both the lectern and wireless microphones into the room.

Room Layout for Conferencing System



The block diagram of this system is shown in the following figure. The channel names are labeled with the virtual channel names that are created by default by the SoundStructure Studio software.

Block Diagram of Conferencing System


Reinforcement of Presenter Microphones
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
Wireless Mic
Lectern Mic
Program Audio (L)
Program Audio (R)
Phone In
SoundStructure C12 and TEL1
Amplifier 1
Amplifier 2
Amplifier 3
Phone Out

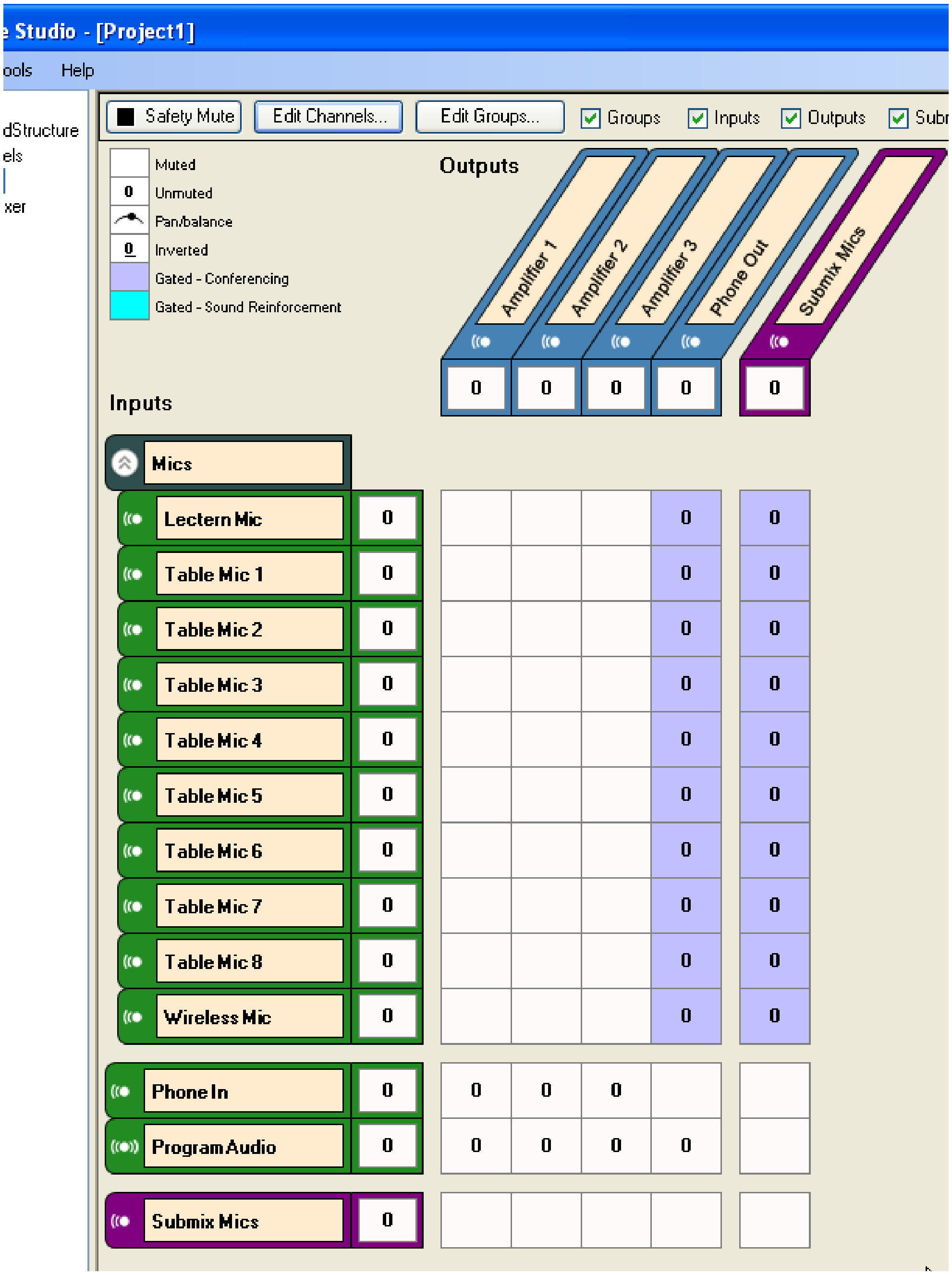
SoundStructure Studio Steps

Creating the design described in the previous section will require a SoundStructure C12 and single line telephony solution.

Matrix Settings

The matrix that is created by SoundStructure Studio is shown in the following figure.

Project Matrix Settings


Studio - [Project1]
ools
Help
dStructure
els
xer
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Subr
Muted
Unmuted
Pan/balance
Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs

Inputs		Amplifier 1	Amplifier 2	Amplifier 3	Phone Out	Submix Mics
		0	0	0	0	0
Mics						
Lectern Mic	0				0	0
Table Mic 1	0				0	0
Table Mic 2	0				0	0
Table Mic 3	0				0	0
Table Mic 4	0				0	0
Table Mic 5	0				0	0
Table Mic 6	0				0	0
Table Mic 7	0				0	0
Table Mic 8	0				0	0
Wireless Mic	0				0	0
Phone In	0	0	0	0		
Program Audio	0	0	0	0	0	
Submix Mics	0					

To add the reinforcement of the wireless and lectern microphones, the lectern microphone will only be reinforced into Amplifier zones 2 and 3 and not in Amplifier zone 1. Because the wireless microphone may be in any zone, it is reinforced into all zones.

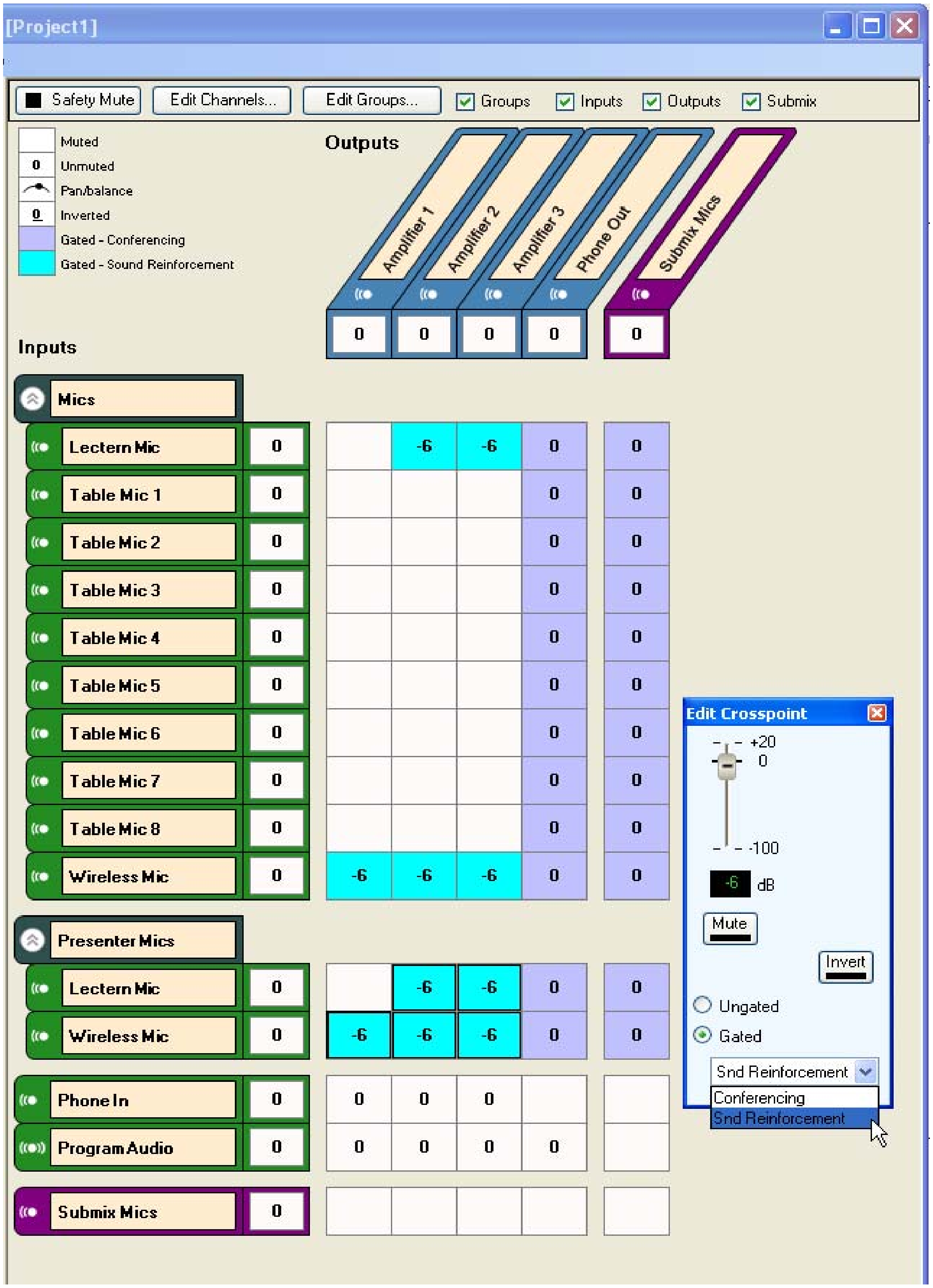
To simplify the system, a presenter group will be created and the wireless microphone and lectern mic will be added to the group. The wireless and lectern microphone can remain in the “Mics” group so that all the microphones may be muted by simply sending a mute command to the “Mics” group.

In addition the multiple matrix crosspoints of the reinforced mics can be selected, and at one time, the value set to -6dB and the *Snd Reinforcement* version of the input processing selected. This will result in the light blue background for the reinforced crosspoints. The reinforcement level can be adjusted if, for instance, the lectern microphone needs to be reinforced at a louder level to the rear of the room.

All microphones are sent to the remote telephony participant as shown with the routing of the conferencing version of the microphones to the “Phone Out” virtual channel.

The resulting matrix will look like the following figure.

Project Matrix Page


[Project1]
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Inputs
Edit Crosspoint
+20
0
-100
Mute
Invert

Inputs	Level	Amplifier 1	Amplifier 2	Amplifier 3	Phone Out	Submix Mics
Output level		0	0	0	0	0
Mics						
Lectern Mic	0		-6	-6	0	0
Table Mic 1	0				0	0
Table Mic 2	0				0	0
Table Mic 3	0				0	0
Table Mic 4	0				0	0
Table Mic 5	0				0	0
Table Mic 6	0				0	0
Table Mic 7	0				0	0
Table Mic 8	0				0	0
Wireless Mic	0	-6	-6	-6	0	0
Presenter Mics						
Lectern Mic	0		-6	-6	0	0
Wireless Mic	0	-6	-6	-6	0	0
Phone In	0	0	0	0		
Program Audio	0	0	0	0	0	
Submix Mics	0					

Edit Crosspoint: -6 dB

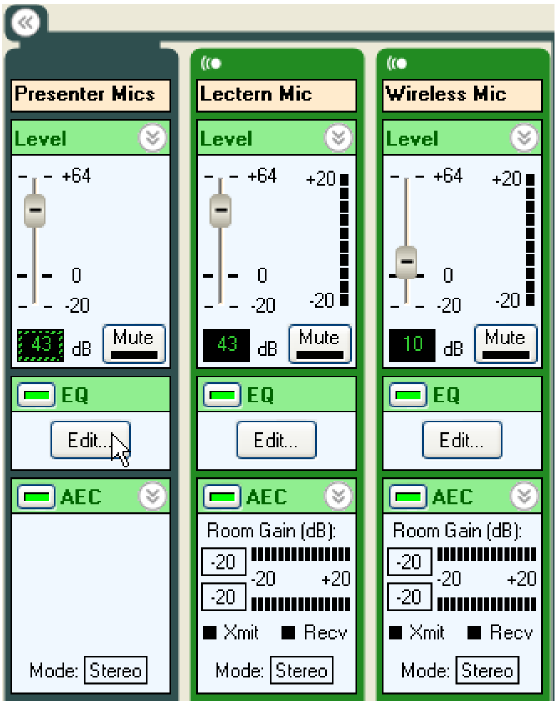
○ Ungated
● Gated

Snd Reinforcement (options: Conferencing, Snd Reinforcement)

Channels Settings

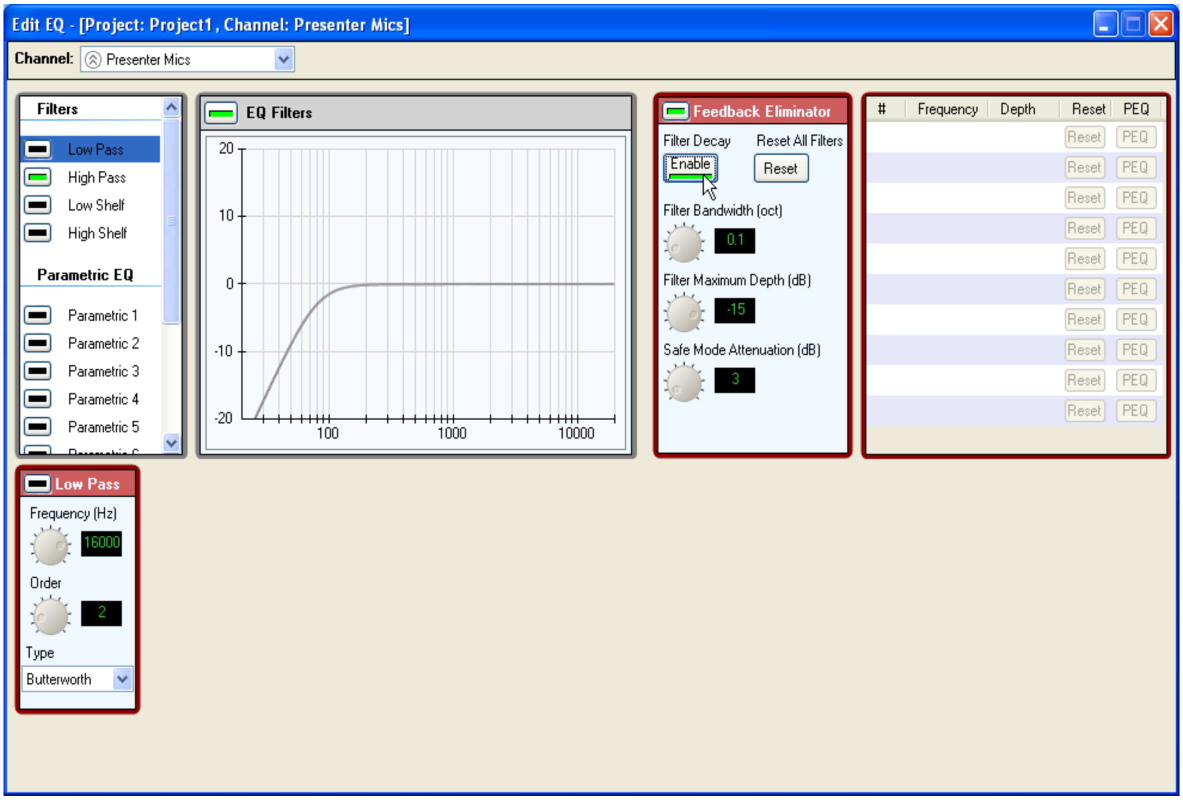
The next step is to enable the feedback processing on the wireless and lectern microphone. This can be done from the channels page by clicking on the EQ button for the “Presenter Mics” group as shown in the following figure.

Project Channels Settings


Presenter Mics
Lectern Mic
Wireless Mic
Level
+64
0
-20
+20
43
dB
10
Mute
EQ
Edit...
AEC
Room Gain (dB):
Xmit
Recv
Mode:
Stereo

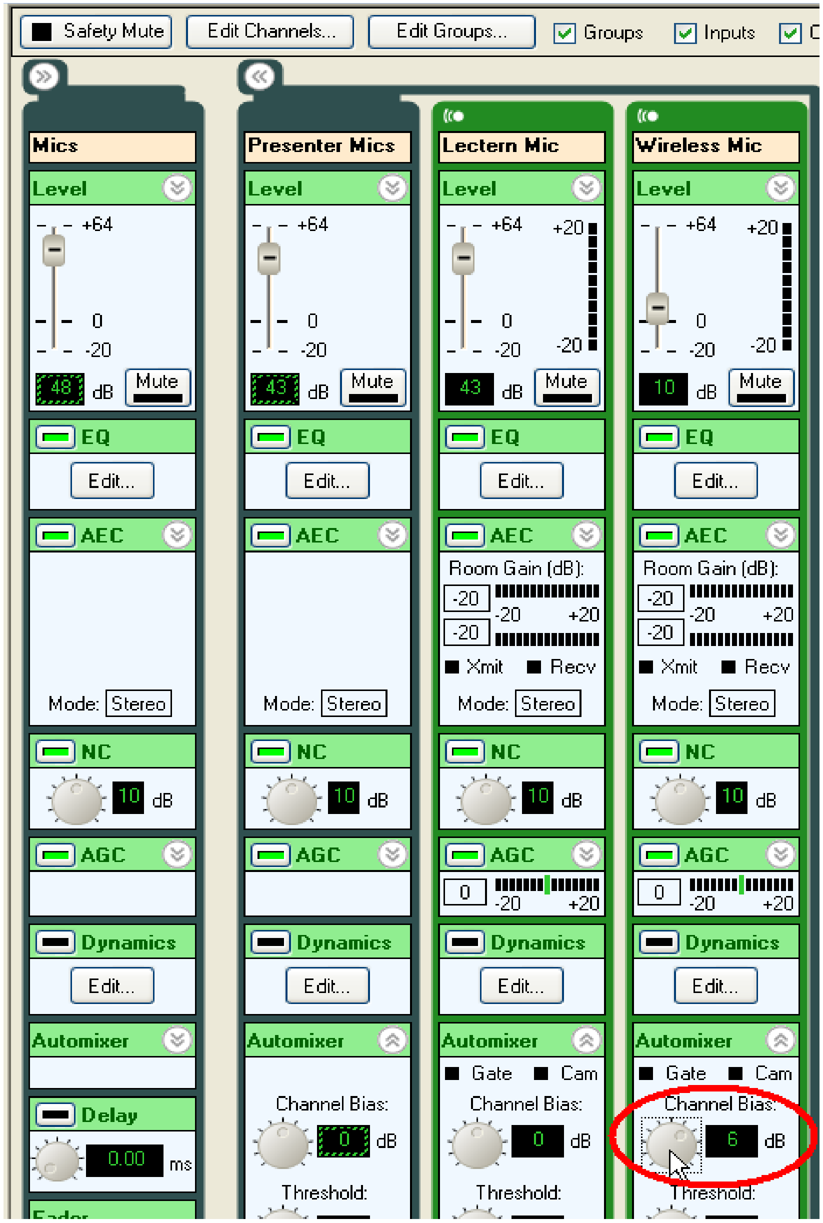
The feedback processing is enabled for the “Presenter Mics” by clicking the enable button next to the Feedback Eliminator name. In addition the Filter Decay feature can be enabled as shown in the next figure.

Project Feedback Processing


Edit EQ - [Project: Project1 , Channel: Presenter Mics]
Channel: Presenter Mics
Filters
Low Pass
High Pass
Low Shelf
High Shelf
Parametric EQ
Parametric 1
Parametric 2
Parametric 3
Parametric 4
Parametric 5
EQ Filters
20
10
0
-10
-20
100
1000
10000
Feedback Eliminator
Filter Decay
Enable
Reset All Filters
Reset
Filter Bandwidth (oct)
0.1
Filter Maximum Depth (dB)
-15
Safe Mode Attenuation (dB)
3
#
Frequency
Depth
Reset
PEQ
Low Pass
Frequency (Hz)
16000
Order
2
Type
Butterworth

To ensure the wireless microphone will be the active microphone if the presenter with the wireless microphone is picked up by another nearby microphone, the automixer channel bias for the wireless microphone will be set to 6dB as shown in the following figure.

Automixer Channel for Wireless Microphones


Safety Mute
Edit Channels...
Edit Groups...
Groups
Inputs
Mics
Presenter Mics
Lectern Mic
Wireless Mic
Level
+64
0
-20
+20
dB
Mute
EQ
Edit...
AEC
Room Gain (dB):
Xmit
Recv
NC
AGC
Dynamics
Automixer
Gate
Cam
Delay
Fader

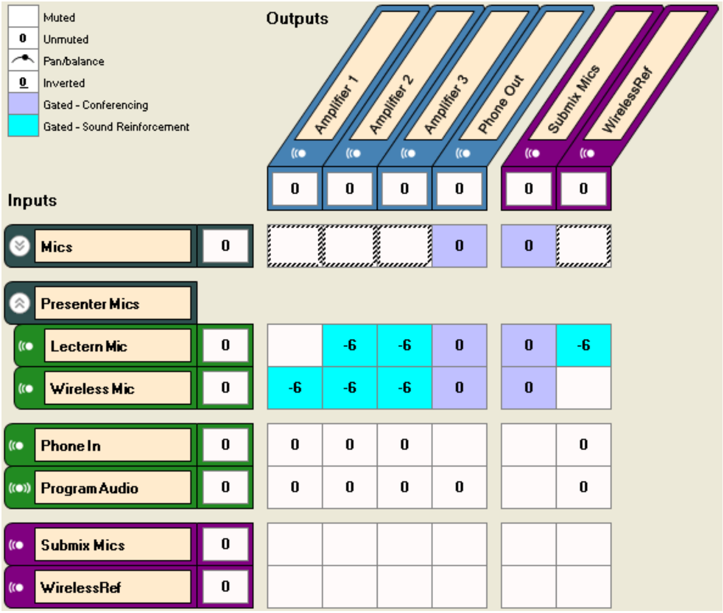
Mode: Stereo

Channel Bias: 6 dB

Threshold:

Finally, it is necessary to review the AEC reference for the different microphones to ensure that acoustic echoes are canceled in the system. The AEC reference for the wireless microphone should include the lectern microphone (as that will be reinforced into the room) and any remote audio sources - the phone line in this case, and the program audio material.

The first step to creating the wireless microphone’s reference is to build this reference by creating a new submix called “WirelessRef” as shown in the following figure.



☐ Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement

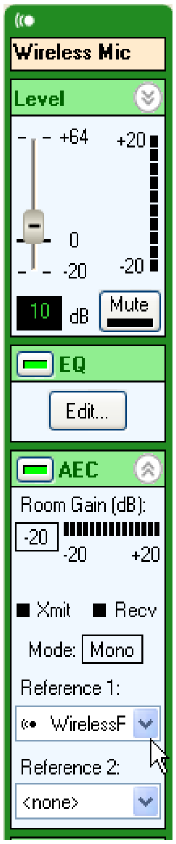
Outputs

Inputs

		Amplifier 1	Amplifier 2	Amplifier 3	Phone Out	Submix Mics	WirelessRef
		0	0	0	0	0	0
Mics	0				0	0	
Presenter Mics							
Lectern Mic	0		-6	-6	0	0	-6
Wireless Mic	0	-6	-6	-6	0	0	
Phone In	0	0	0	0			0
Program Audio	0	0	0	0	0		0
Submix Mics	0						
WirelessRef	0						

The AEC reference for the wireless microphone is assigned to the new submix as shown in the next figure.

AEC Reference for Wireless Microphone


Wireless Mic
Level
+64
0
-20
+20
-20
Mute
EQ
Edit...
AEC
-20
+20
Xmit
Recv

10 dB

Room Gain (dB): -20

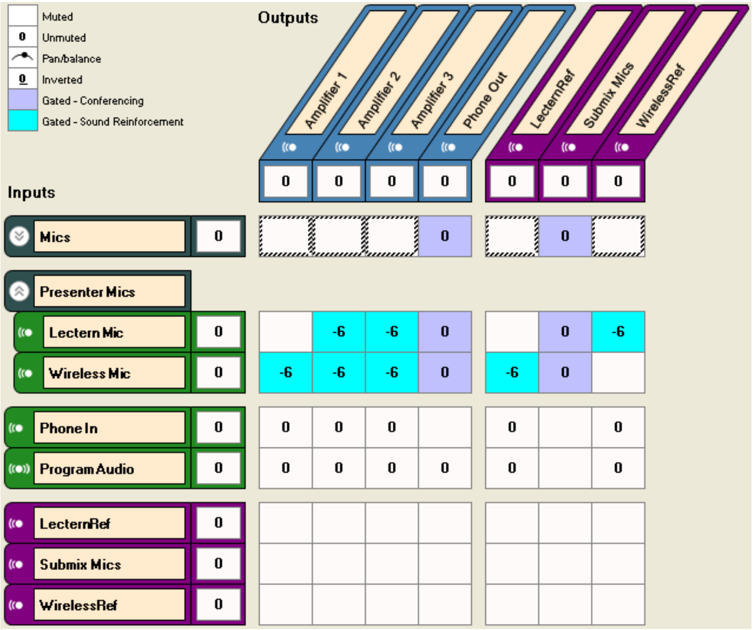
Mode: Mono

Reference 1: WirelessF

Reference 2: <none>

The same approach can be taken with the lectern microphone, creating a submix called “LecternRef” that includes the reinforced wireless microphone, the remote audio sources, and the program audio. The new matrix will appear as shown in the following figure.

Project Matrix Page

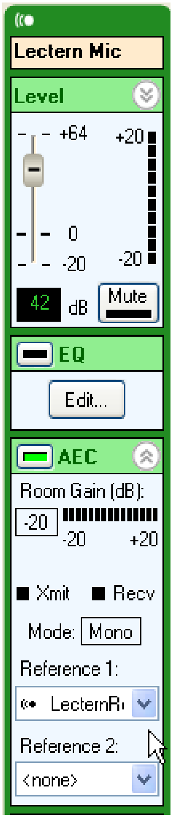


☐ Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement

Inputs		Amplifier 1	Amplifier 2	Amplifier 3	Phone Out	LecternRef	Submix Mics	WirelessRef
Outputs		0	0	0	0	0	0	0
Mics	0	▨	▨	▨	0	▨	0	▨
Presenter Mics								
Lectern Mic	0		-6	-6	0		0	-6
Wireless Mic	0	-6	-6	-6	0	-6	0	
Phone In	0	0	0	0		0		0
Program Audio	0	0	0	0	0	0		0
LecternRef	0							
Submix Mics	0							
WirelessRef	0							

The AEC reference for the lectern mic can then be set to the “LecternRef” submix as shown in the next figure.

.AEC Reference for Lectern Microphone


Lectern Mic

Level

42 dB

Room Gain (dB): -20

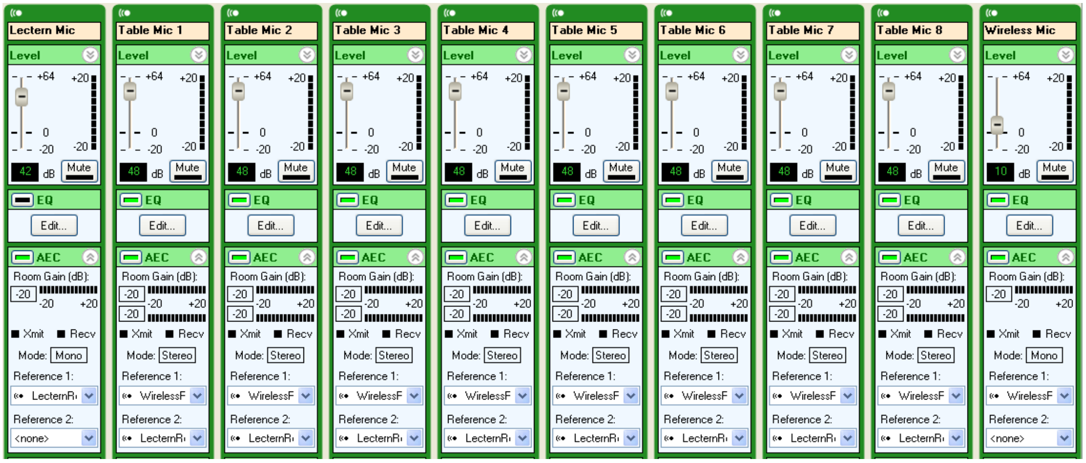
Mode: Mono

Reference 1: LecternR

Reference 2: <none>

Finally, the reference for the table microphones can be set to include both the lectern and wireless microphone references. Since two references can be configured per microphone, the first reference will be set to “WirelessRef” and the second reference will be set to “LecternRef”.

Microphone References



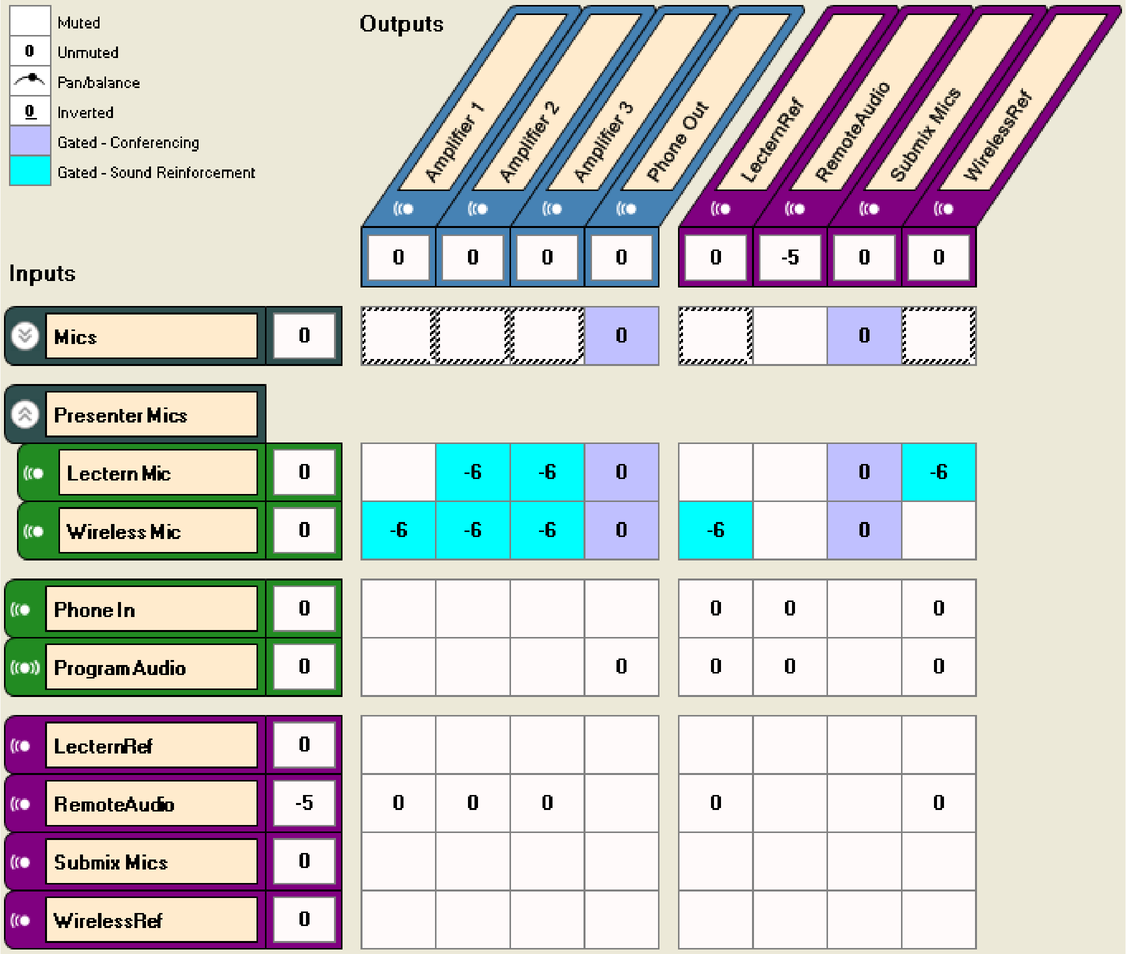
	Lectern Mic	Table Mic 1	Table Mic 2	Table Mic 3	Table Mic 4	Table Mic 5	Table Mic 6	Table Mic 7	Table Mic 8	Wireless Mic
Level (dB)	42	48	48	48	48	48	48	48	48	10
Room Gain (dB):	-20	-20 -20	-20 -20	-20 -20	-20 -20	-20 -20	-20 -20	-20 -20	-20 -20	-20
Mode:	Mono	Stereo	Stereo	Stereo	Stereo	Stereo	Stereo	Stereo	Stereo	Mono
Reference 1:	LecternR	WirelessF	WirelessF	WirelessF	WirelessF	WirelessF	WirelessF	WirelessF	WirelessF	WirelessF
Reference 2:	<none>	LecternR	LecternR	LecternR	LecternR	LecternR	LecternR	LecternR	LecternR	<none>

To further simplify the system with an eye towards in-room volume control, another submix “RemoteAudio” will be created that contains just the remote audio signals - the telephone and the program audio. This way the in-room volume control can adjust the “RemoteAudio” submix to increase or decrease the level of the remote audio into the local room. See the following figure for how the new matrix will appear.

Keep in mind that the “RemoteAudio” channel should not be sent to the “Phone Out” signal to prevent the “Phone In” channel from being routed to the “Phone Out” signal causing a persistent electronic echo of the telephone talker back to the telephone talker.

The “RemoteAudio” submix will also be routed to the different amplifier zones and remote telephone participants.

Project Matrix Page



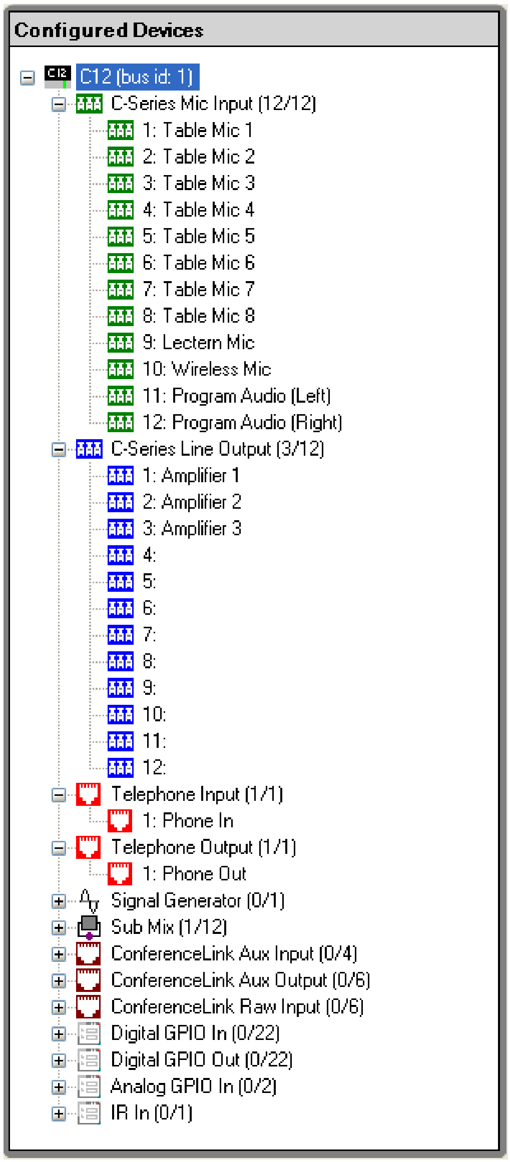
	Muted
0	Unmuted
	Pan/balance
0	Inverted
	Gated - Conferencing
	Gated - Sound Reinforcement

Inputs		Amplifier 1	Amplifier 2	Amplifier 3	Phone Out	LecternRef	RemoteAudio	Submix Mics	WirelessRef
Outputs		0	0	0	0	0	-5	0	0
Mics	0				0			0	
Presenter Mics									
Lectern Mic	0		-6	-6	0			0	-6
Wireless Mic	0	-6	-6	-6	0	-6		0	
Phone In	0					0	0		0
Program Audio	0				0	0	0		0
LecternRef	0								
RemoteAudio	-5	0	0	0		0			0
Submix Mics	0								
WirelessRef	0								

Wiring Information

The system should be wired according to the information found in the wiring page and shown in the next figure. To wire the system with virtual channels on other physical inputs or outputs, simply drag the channels to their desired locations and then wire the system according to the modified wiring information.

Project Wiring Information



Configured Devices

- C12 (bus id: 1)
  - C-Series Mic Input (12/12)
    - 1: Table Mic 1
    - 2: Table Mic 2
    - 3: Table Mic 3
    - 4: Table Mic 4
    - 5: Table Mic 5
    - 6: Table Mic 6
    - 7: Table Mic 7
    - 8: Table Mic 8
    - 9: Lectern Mic
    - 10: Wireless Mic
    - 11: Program Audio (Left)
    - 12: Program Audio (Right)
  - C-Series Line Output (3/12)
    - 1: Amplifier 1
    - 2: Amplifier 2
    - 3: Amplifier 3
    - 4:
    - 5:
    - 6:
    - 7:
    - 8:
    - 9:
    - 10:
    - 11:
    - 12:
  - Telephone Input (1/1)
    - 1: Phone In
  - Telephone Output (1/1)
    - 1: Phone Out
  - Signal Generator (0/1)
  - Sub Mix (1/12)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

Controlling The System

The presenter microphones can be muted by sending the mute command to the “Presenter Mics” group with the following command.

```
set mute “Presenter Mics” 1
```

and may be unmuted by sending the command

```
set mute “Presenter Mics” 0
```

The reinforcement of the wireless microphone may be disabled by muting the reinforced crosspoints as shown next.

```
set matrix_mute “Wireless Mic” “Amplifier 1” 1
set matrix_mute “Wireless Mic” “Amplifier 2” 1
set matrix_mute “Wireless Mic” “Amplifier 3” 1
```

The reinforcement of the wireless microphone may be enabled by setting the mute status to 0.

```
set matrix_mute “Wireless Mic” “Amplifier 1” 0
set matrix_mute “Wireless Mic” “Amplifier 2” 0
set matrix_mute “Wireless Mic” “Amplifier 3” 0
```

The amount of reinforcement of the “Wireless Mic” channel to the zone 1 amplifier can be increased and decreased, respectively, by 1dB with the following commands.

```
inc matrix_gain “Wireless Mic” “Amplifier 1” 1
dec matrix_gain “Wireless Mic” “Amplifier 1” 1
```

It is also possible to set user minimum and maximum values for the crosspoint levels to prevent adding too much gain for reinforcement. The maximum crosspoint gain settings can be set to -3dB for the wireless microphone to zone 1 amplifier with the following command.

```
set matrix_gain max “Wireless Mic” “Amplifier 1” -3
```

When the volume of the crosspoint is raised, the value will not become larger than -3dB.

The remote audio being played into all the zones can be controlled by using the “RemoteAudio” submix. In room volume may be increased with the following volume command.

```
inc fader “RemoteAudio” 1
```

and in room volume of the remote participants may be reduced with the following command.

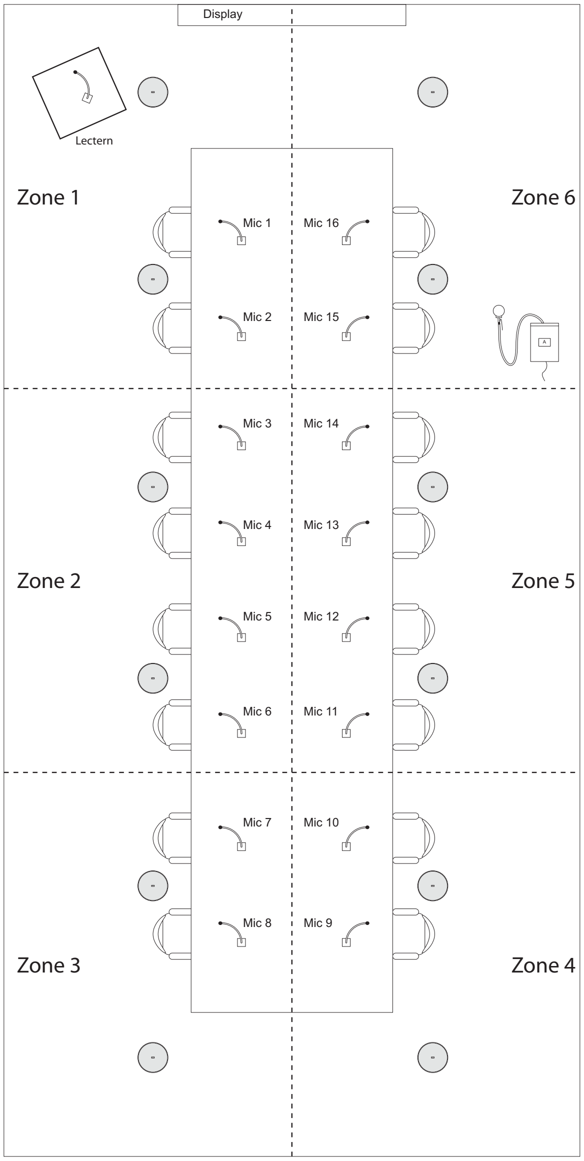
```
dec fader “RemoteAudio” 1
```

Creating a Sixteen Microphones with Six-Zone Sound Reinforcement Conferencing System

This example shows how to use the sound reinforcement and conferencing processing to create an audio conferencing solution that includes a lectern microphone, wireless microphone, and sixteen tabletop microphones that are reinforced into the room. This example includes sixteen table microphones, a lectern microphone, a wireless microphone, stereo program audio, a single telephony interface, and six zones of audio amplifiers for reinforcement.

The layout for this style of room can be seen in the following figure along with the zone definitions. In this room, the lectern microphone will be reinforced into zones 2-6, the wireless microphone reinforced into zones 1-6, and each table microphone zone reinforced into all the other zones at varying levels depending on the proximity between zones.

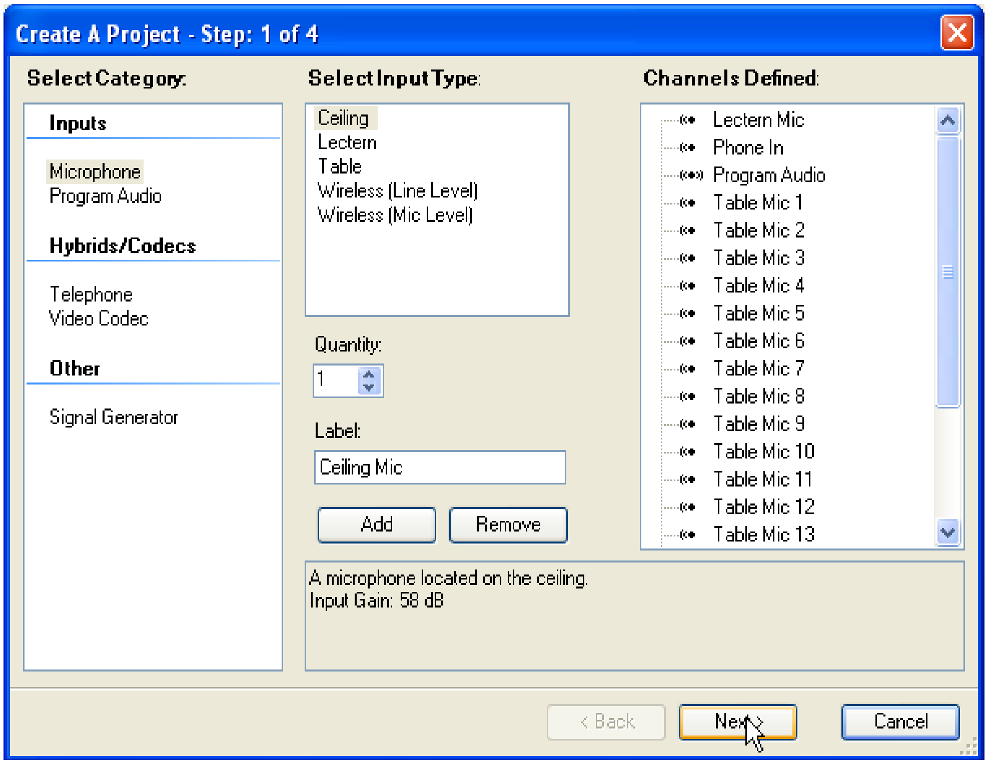
Room Layout for Conferencing System


Display
Lectern
Zone 1
Zone 6
Zone 2
Zone 5
Zone 3
Zone 4
Mic 1
Mic 2
Mic 3
Mic 4
Mic 5
Mic 6
Mic 7
Mic 8
Mic 9
Mic 10
Mic 11
Mic 12
Mic 13
Mic 14
Mic 15
Mic 16

SoundStructure Studio Steps

Step 1 - Select Inputs

The system is designed with 16 table microphones, one lectern mic, one wireless mic with line level input, one stereo VSX8000 video codec, and a single telephony interface.


Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Telephone
Video Codec
Other
Signal Generator
Select Input Type:
Ceiling
Lectern
Table
Wireless (Line Level)
Wireless (Mic Level)
Channels Defined:
Lectern Mic
Phone In
Program Audio
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Table Mic 5
Table Mic 6
Table Mic 7
Table Mic 8
Table Mic 9
Table Mic 10
Table Mic 11
Table Mic 12
Table Mic 13
Add
Remove
< Back
Next >
Cancel

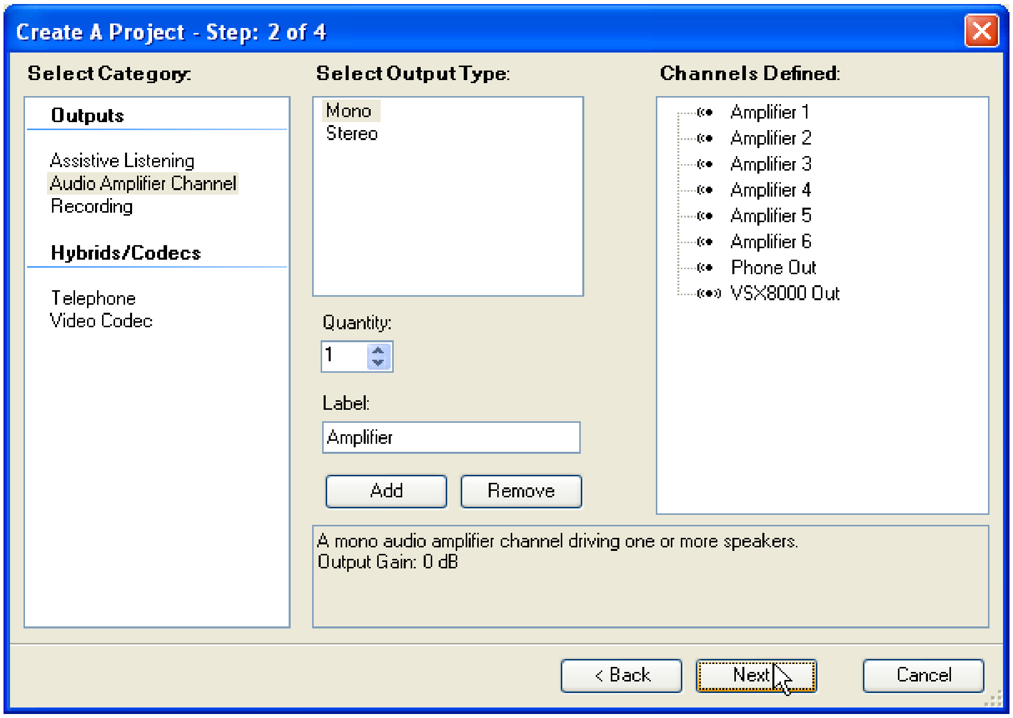
Quantity: 1

Label: Ceiling Mic

A microphone located on the ceiling.
Input Gain: 58 dB

Step 2 - Select Outputs

Six mono audio amplifiers are added to the system in this step. The output to the telephone line and VSX8000 were created when their respective input components were added to the system in step 1.


Create A Project - Step: 2 of 4
Select Category:
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Telephone
Video Codec
Select Output Type:
Mono
Stereo
Channels Defined:
Amplifier 1
Amplifier 2
Amplifier 3
Amplifier 4
Amplifier 5
Amplifier 6
Phone Out
VSX8000 Out
Add
Remove
< Back
Next
Cancel

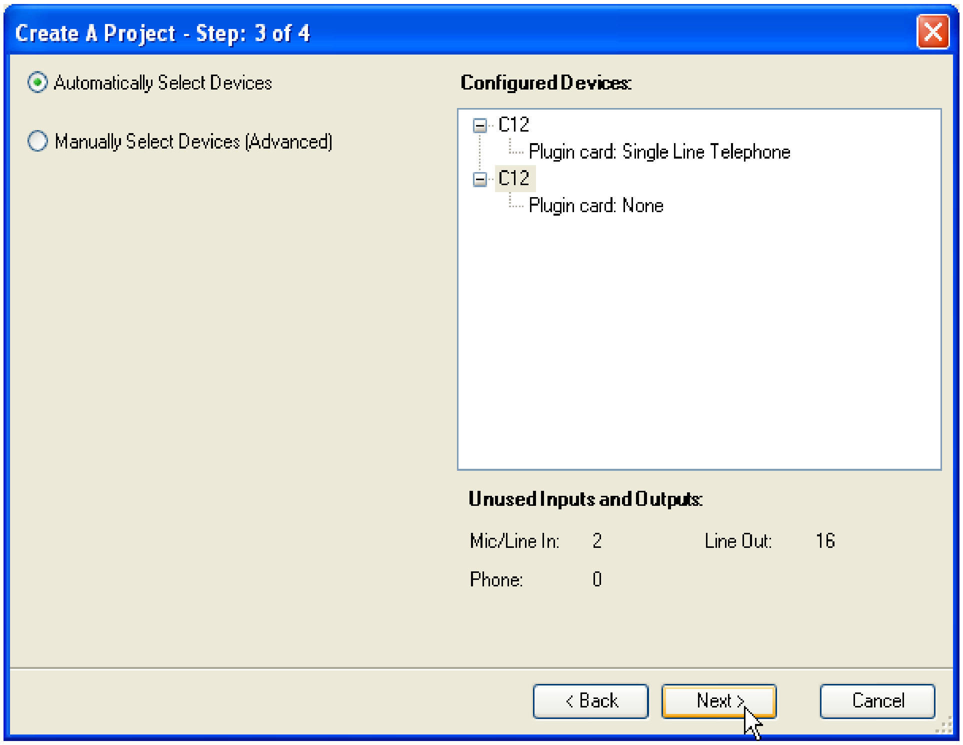
Quantity: 1

Label: Amplifier

A mono audio amplifier channel driving one or more speakers.
Output Gain: 0 dB

Step 3 - Select Equipment

The default equipment selection will use two C12's, and a TEL1 telephony card.



Create A Project - Step: 3 of 4

● Automatically Select Devices

○ Manually Select Devices (Advanced)

Configured Devices:

- C12
  - Plugin card: Single Line Telephone
- C12
  - Plugin card: None

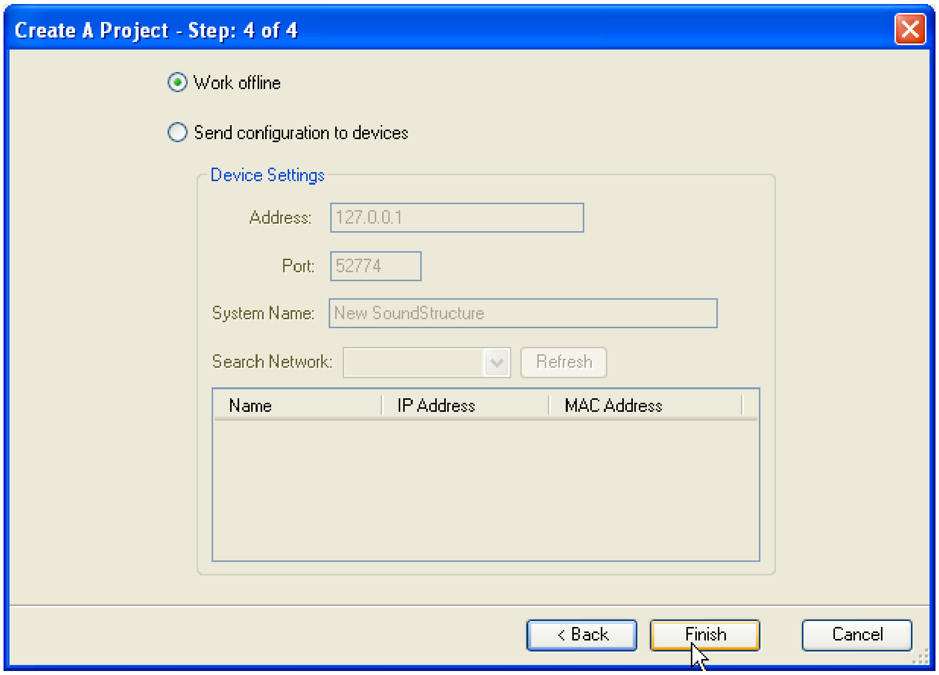
Unused Inputs and Outputs:

Mic/Line In:	2	Line Out:	16
Phone:	0		

< Back
Next >
Cancel

Step 4 - Work Offline Or Online

As there are many matrix settings to change, we'll work off line and adjust the crosspoints.



Create A Project - Step: 4 of 4

● Work offline

○ Send configuration to devices

Device Settings

Address: 127.0.0.1

Port: 52774

System Name: New SoundStructure

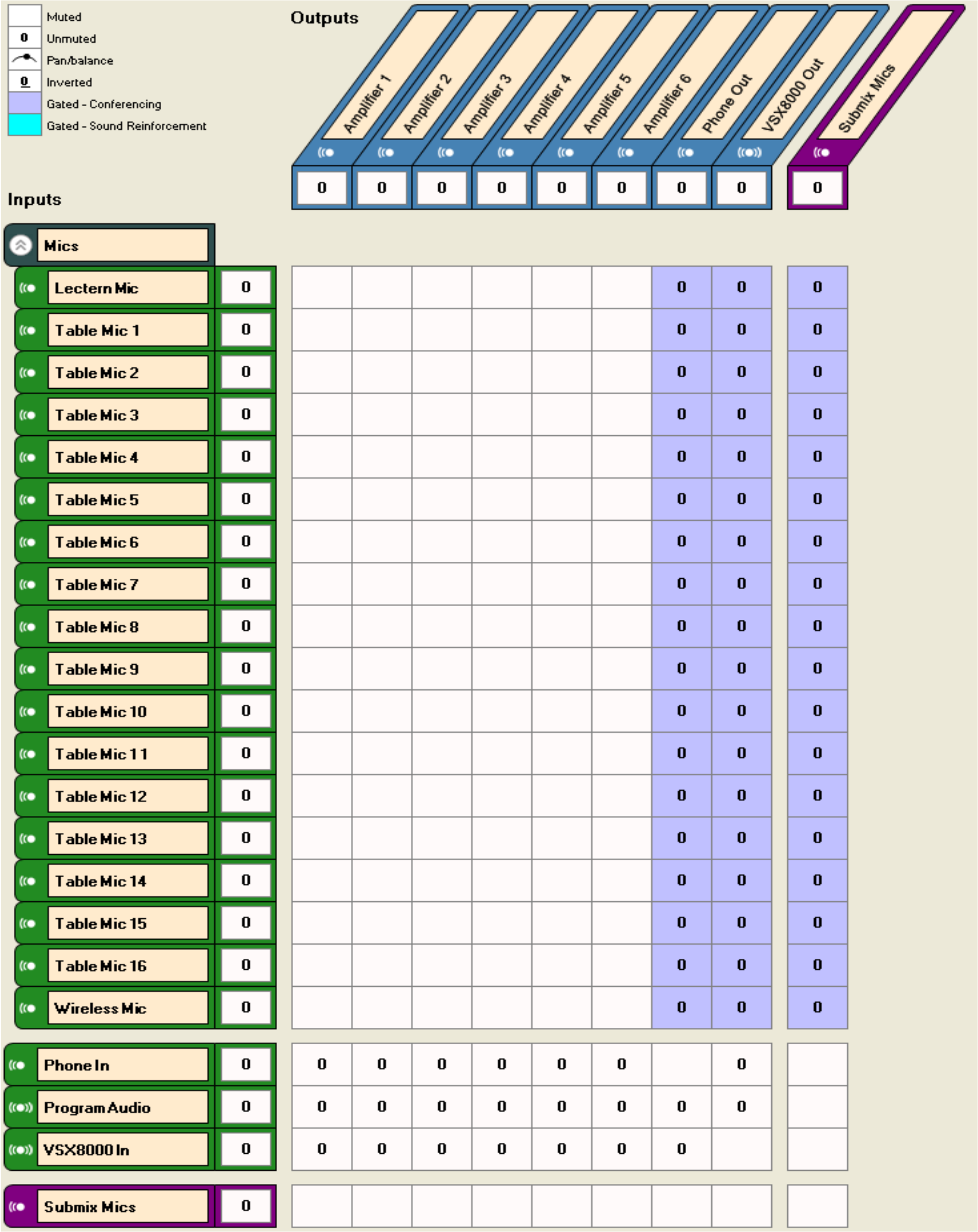
Search Network:

Refresh
Name
IP Address
MAC Address
< Back
Finish
Cancel

Matrix Settings

The default matrix with the desired inputs and outputs is shown in the following figure.

Default Matrix Settings



☐	Muted
0	Unmuted
⌒	Pan/balance
0	Inverted
🟪	Gated - Conferencing
🟦	Gated - Sound Reinforcement

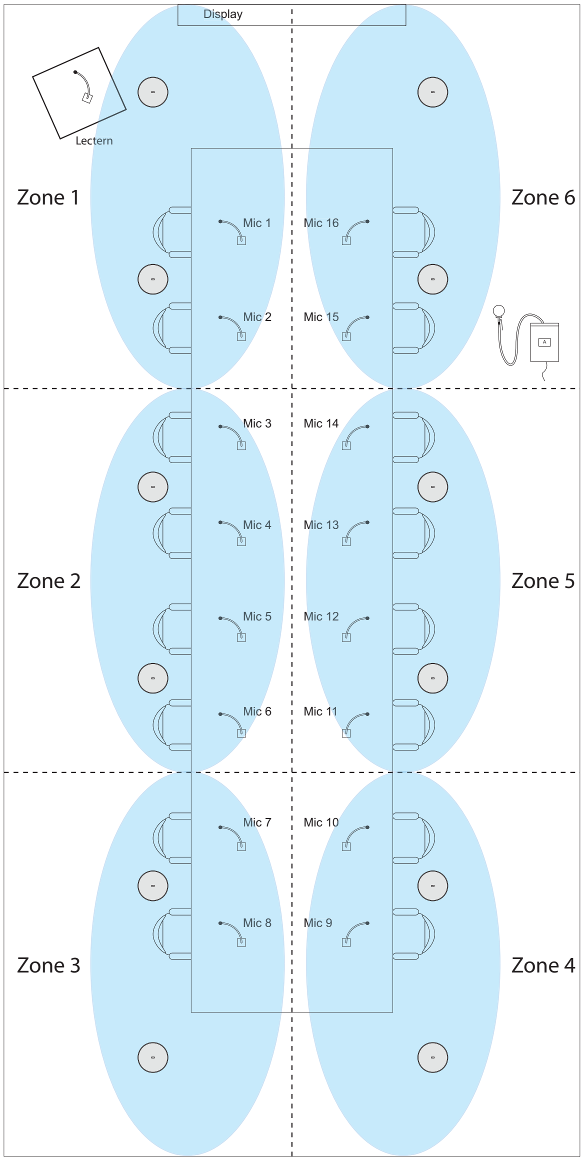
Outputs

Inputs

Inputs		Amplifier 1	Amplifier 2	Amplifier 3	Amplifier 4	Amplifier 5	Amplifier 6	Phone Out	VSX8000 Out	Submix Mics
		0	0	0	0	0	0	0	0	0
Mics										
Lectern Mic	0							0	0	0
Table Mic 1	0							0	0	0
Table Mic 2	0							0	0	0
Table Mic 3	0							0	0	0
Table Mic 4	0							0	0	0
Table Mic 5	0							0	0	0
Table Mic 6	0							0	0	0
Table Mic 7	0							0	0	0
Table Mic 8	0							0	0	0
Table Mic 9	0							0	0	0
Table Mic 10	0							0	0	0
Table Mic 11	0							0	0	0
Table Mic 12	0							0	0	0
Table Mic 13	0							0	0	0
Table Mic 14	0							0	0	0
Table Mic 15	0							0	0	0
Table Mic 16	0							0	0	0
Wireless Mic	0							0	0	0
Phone In	0	0	0	0	0	0	0		0	
Program Audio	0	0	0	0	0	0	0	0	0	
VSX8000 In	0	0	0	0	0	0	0	0		
Submix Mics	0									

The next step is to create the microphone zone groups that will simplify setting up the reinforcement levels. The designed zones are shown in the following figure.

Microphone Zone Groups


Display
Lectern
Zone 1
Zone 6
Mic 1
Mic 16
Mic 2
Mic 15
Mic 3
Mic 14
Mic 4
Mic 13
Zone 2
Zone 5
Mic 5
Mic 12
Mic 6
Mic 11
Mic 7
Mic 10
Mic 8
Mic 9
Zone 3
Zone 4

As part of the design process, the appropriate reinforcement levels would be determined and a mapping similar to the one shown in the following figure would be created as the baseline reinforcement in the room. This mapping shows how the different input zones will be mapped to the different amplifier zones. For example, the zone 1 microphones are mapped to zones 2, 3, 4, 5, and 6 with a gain of -9, -6, -6, -9, and -12dB respectively. The zone numbering matches the room layout description.

Input Mapping

Amplifier

1	**2**	**3**	**4**	**5**	**6**

Input Mapping

	Lectern		**-6**	**-6**	**-6**	**-6**	**-6**
	Wireless	**-6**	**-6**	**-6**	**-6**	**-6**	**-6**
	1	▨		**-6**	**-6**	**-9**	**-12**
	2	**-9**	▨		**-9**	**-12**	**-9**
Zone	**3**	**-6**	**-9**	▨	**-12**	**-9**	**-6**
	4	**-6**	**-9**	**-12**	▨	**-12**	**-6**
	5	**-9**	**-12**	**-9**	**-12**	▨	**-12**

To create a zoned reinforcement system with the reinforcement levels shown in the table, the matrix crosspoints for the zones must be adjusted to match the designed reinforcement matrix. The first step is to create the zone groups and then map the zone groups to the amplifier outputs with the desired crosspoints and sound reinforcement version of the input processing.

To create the different zones, select the **Edit Groups...** button and follow the instructions in the section Creating Virtual Channel Groups in Customizing SoundStructure Designs. The result should be six zones of microphones that include the microphones shown in the drawing of the room. Once the zones have been created into virtual channel groups, the groups may be collapsed so that the matrix operates at the group level - hiding the detail of the underlying microphones as shown in the following figure.

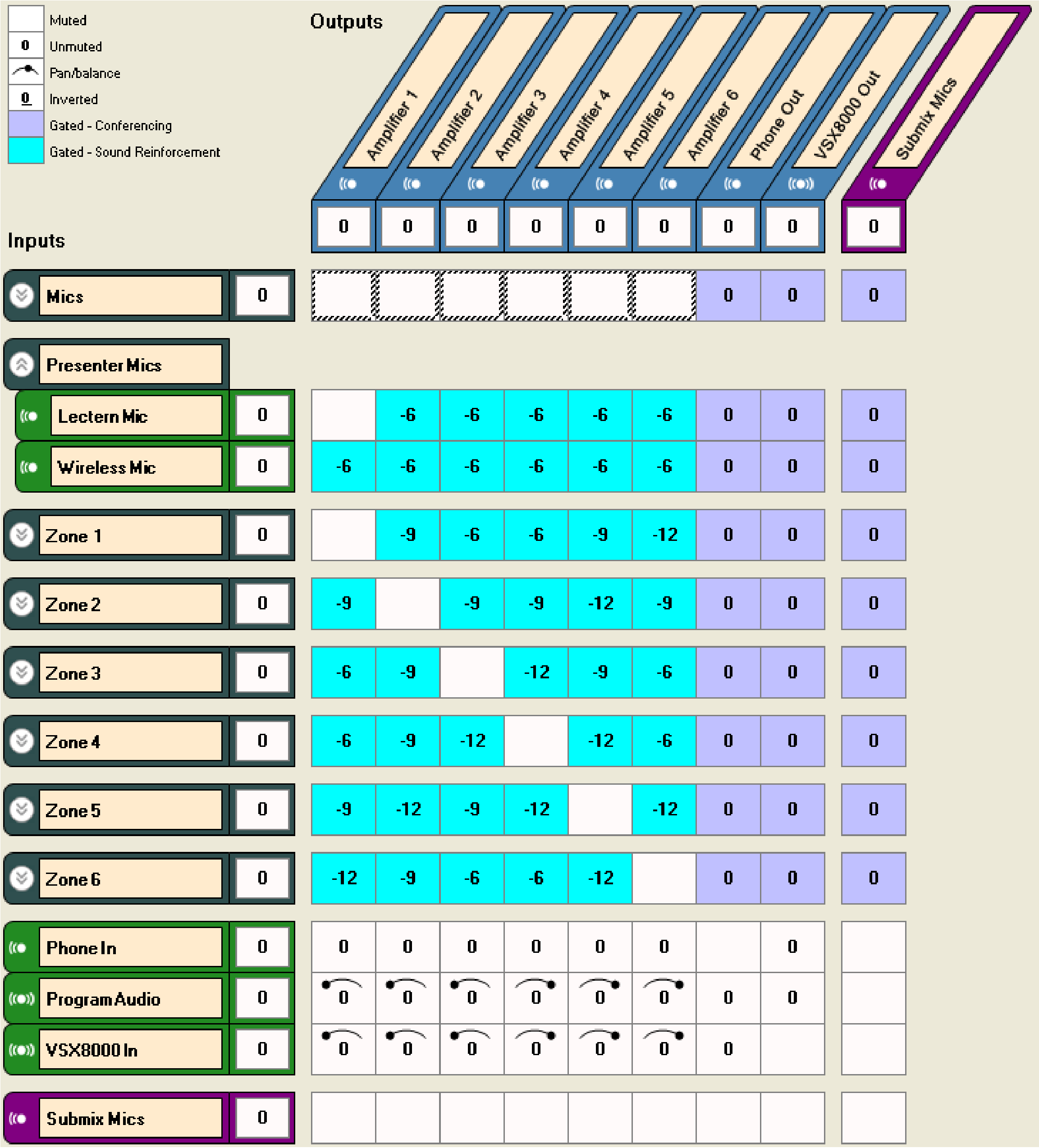
In this example Zone 1 includes the microphones shown in the following table.

Microphones Included in Zone 1

Zone	Microphones
Zone 1	1 and 2
Zone 2	3, 4, 5, and 6
Zone 3	7 and 8
Zone 4	9 and 10
Zone 5	11, 12, 13, and 14
Zone 6	15 and 16

The next step is to map the stereo program audio and video codec audio to the appropriate left and right loudspeakers in the room. The result is shown in the following figure where the left channel of the audio is panned to the amplifiers in zones 1, 2, and 3 and the right channel of the audio is panned to amplifiers 4, 5, and 6.

Stereo Program Audio and Video Codec Audio



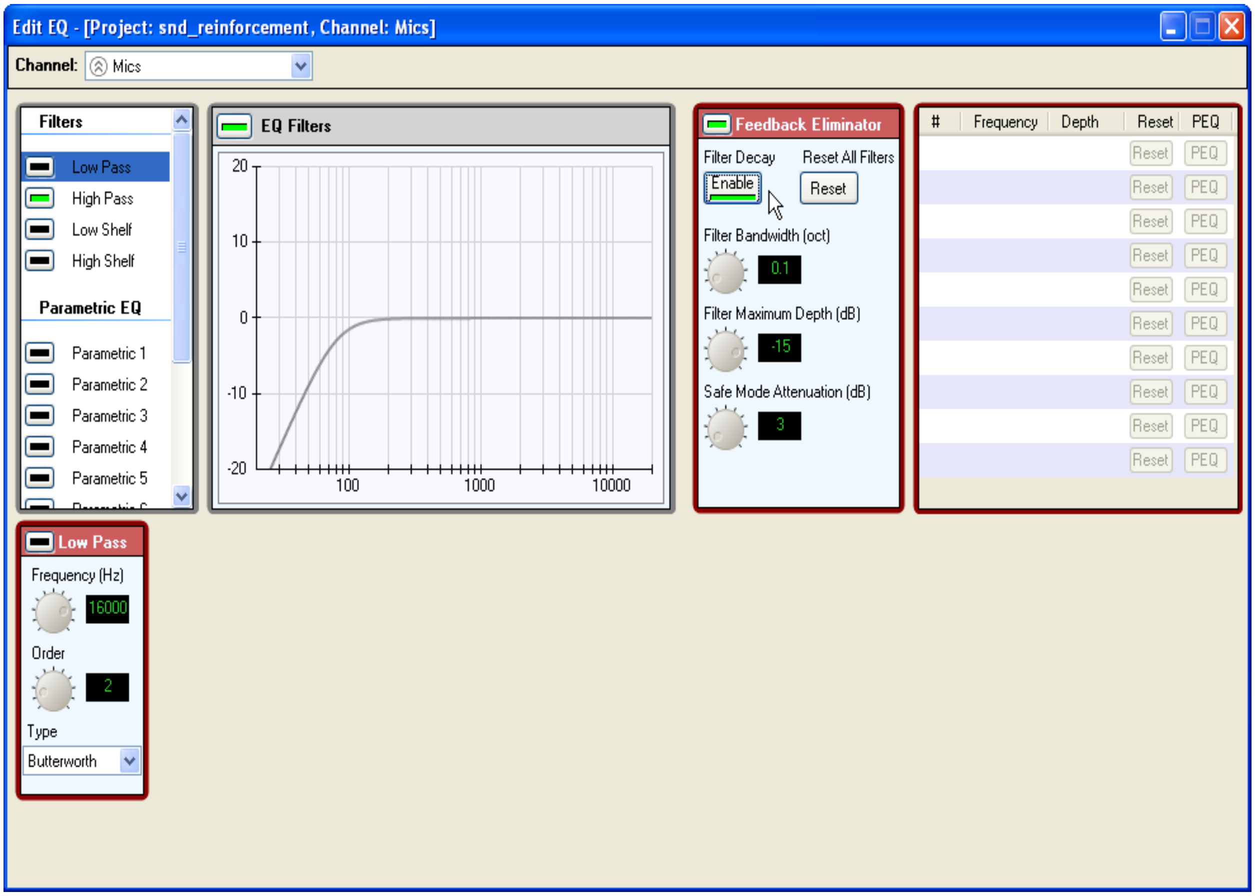
	Muted
0	Unmuted
	Pan/balance
0	Inverted
	Gated - Conferencing
	Gated - Sound Reinforcement

Inputs		Amplifier 1	Amplifier 2	Amplifier 3	Amplifier 4	Amplifier 5	Amplifier 6	Phone Out	VSX8000 Out	Submix Mics
Outputs		0	0	0	0	0	0	0	0	0
Mics	0							0	0	0
Presenter Mics										
Lectern Mic	0		-6	-6	-6	-6	-6	0	0	0
Wireless Mic	0	-6	-6	-6	-6	-6	-6	0	0	0
Zone 1	0		-9	-6	-6	-9	-12	0	0	0
Zone 2	0	-9		-9	-9	-12	-9	0	0	0
Zone 3	0	-6	-9		-12	-9	-6	0	0	0
Zone 4	0	-6	-9	-12		-12	-6	0	0	0
Zone 5	0	-9	-12	-9	-12		-12	0	0	0
Zone 6	0	-12	-9	-6	-6	-12		0	0	0
Phone In	0	0	0	0	0	0	0		0	
Program Audio	0	0	0	0	0	0	0	0	0	
VSX8000 In	0	0	0	0	0	0	0	0		
Submix Mics	0									

Channels Settings

Once the matrix has been configured, the next step is to enable the feedback processing for each microphone. This can be done easily with the channels page editing the EQ settings for the “Mics” group as shown in the following figure. Notice that the channel selection is set to “Mics” - this will ensure the feedback processing is enabled for all microphones in the system.

Equalizer Settings on the Channels Page



Edit EQ - [Project: snd_reinforcement, Channel: Mics]

Channel: Mics

Filter Bandwidth (oct): 0.1

Filter Maximum Depth (dB): -15

Safe Mode Attenuation (dB): 3

Low Pass

Frequency (Hz): 16000

Order: 2

Type: Butterworth

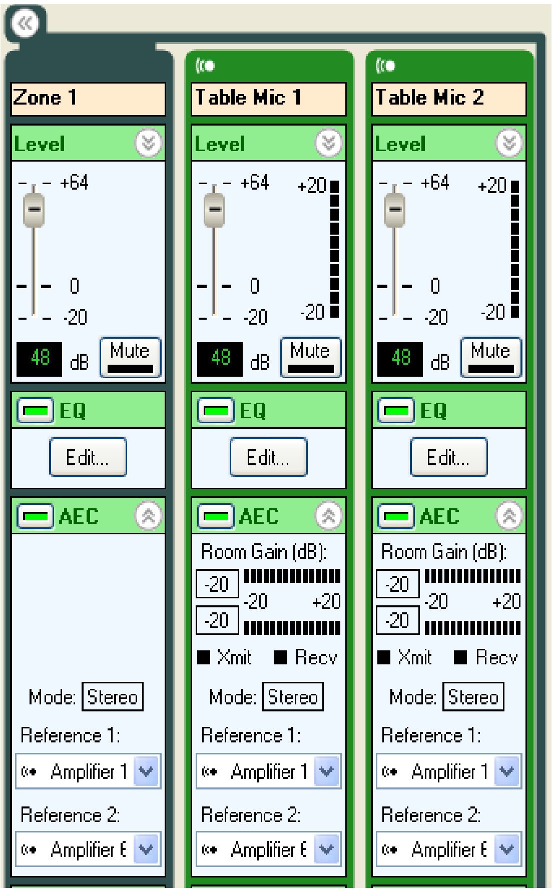
Filters
Low Pass
High Pass
Low Shelf
High Shelf
Parametric EQ
Parametric 1
Parametric 2
Parametric 3
Parametric 4
Parametric 5
EQ Filters
20
10
0
-10
-20
100
1000
10000
Feedback Eliminator
Filter Decay
Enable
Reset All Filters
Reset
#
Frequency
Depth
Reset
PEQ

The next step in the system is to configure the AEC references properly for each microphone.

The table top microphones will have AEC references that are their adjacent left/right zones. For instance Zone 1 and Zone 6 microphones will have Zone 1 and Zone 6 amplifiers selected as their two references, Zone 2 and Zone 5 microphones will have Zone 2 and Zone 5 amplifiers selected as their references, and

Zone 3 and Zone 4 microphones will have Zone 3 and Zone 4 amplifiers selected as shown in the next figure. This figure shows the Zone 1 microphones.

Zone 1 Microphones



Zone 1: Level 48 dB; Mode: Stereo; Reference 1: Amplifier 1; Reference 2: Amplifier E

Table Mic 1: Level 48 dB; Room Gain (dB): -20, -20; Mode: Stereo; Reference 1: Amplifier 1; Reference 2: Amplifier E

Table Mic 2: Level 48 dB; Room Gain (dB): -20, -20; Mode: Stereo; Reference 1: Amplifier 1; Reference 2: Amplifier E

The references for the lectern microphone can also be set to the Zone 1 and Zone 6 amplifiers. The wireless microphone reference should be set to the remote audio, the program audio, and the reinforced audio. This can be done easily by setting the references for the wireless microphones to the Zone 2 and Zone 5 amplifiers.

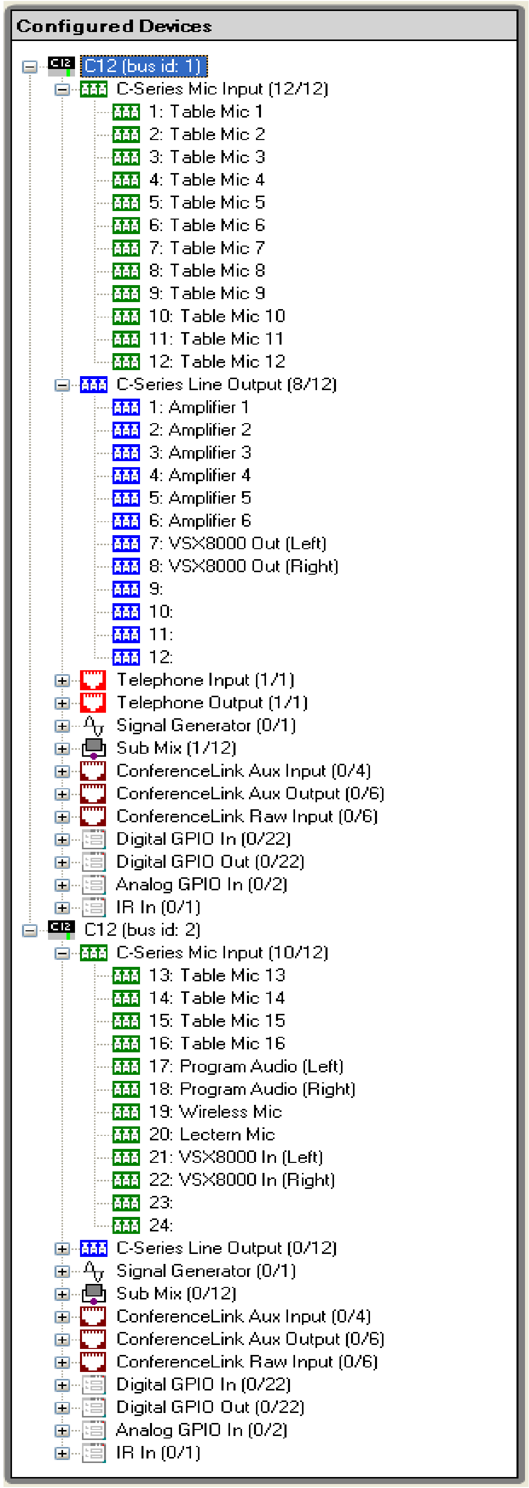
Wiring Information

The system should be wired according to the information found in the wiring page and shown in the following figure. To wire the system with virtual channels on other physical inputs or outputs, simply drag the channels to their desired locations and then wire the system according to the modified wiring information.

In this example, two C12 devices were required to implement the design. The two devices are linked with the OBAM interface and each device is wired as shown in the following figure.

The first C12 is configured to be bus id 1 and the second is configured to be bus id 2 by default assuming the OBAM out of the first device is connected to the OBAM in on the second device.

Project Wiring Information



Configured Devices

- C12 (bus id: 1)
  - C-Series Mic Input (12/12)
    - 1: Table Mic 1
    - 2: Table Mic 2
    - 3: Table Mic 3
    - 4: Table Mic 4
    - 5: Table Mic 5
    - 6: Table Mic 6
    - 7: Table Mic 7
    - 8: Table Mic 8
    - 9: Table Mic 9
    - 10: Table Mic 10
    - 11: Table Mic 11
    - 12: Table Mic 12
  - C-Series Line Output (8/12)
    - 1: Amplifier 1
    - 2: Amplifier 2
    - 3: Amplifier 3
    - 4: Amplifier 4
    - 5: Amplifier 5
    - 6: Amplifier 6
    - 7: VSX8000 Out (Left)
    - 8: VSX8000 Out (Right)
    - 9:
    - 10:
    - 11:
    - 12:
  - Telephone Input (1/1)
  - Telephone Output (1/1)
  - Signal Generator (0/1)
  - Sub Mix (1/12)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)
- C12 (bus id: 2)
  - C-Series Mic Input (10/12)
    - 13: Table Mic 13
    - 14: Table Mic 14
    - 15: Table Mic 15
    - 16: Table Mic 16
    - 17: Program Audio (Left)
    - 18: Program Audio (Right)
    - 19: Wireless Mic
    - 20: Lectern Mic
    - 21: VSX8000 In (Left)
    - 22: VSX8000 In (Right)
    - 23:
    - 24:
  - C-Series Line Output (0/12)
  - Signal Generator (0/1)
  - Sub Mix (0/12)
  - ConferenceLink Aux Input (0/4)
  - ConferenceLink Aux Output (0/6)
  - ConferenceLink Raw Input (0/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

Controlling The System

The system can be controlled in the same manner as the previous examples. The microphones may be muted and unmuted with the following mute commands.

```
set mute “Mics” 1
set mute “Mics” 0
```

The in-room volume for the remote audio may be increased with the fader command on the phone or video codec audio as follows.

```
inc fader “VSX8000 In” 1
inc fader “Phone In” 1
```

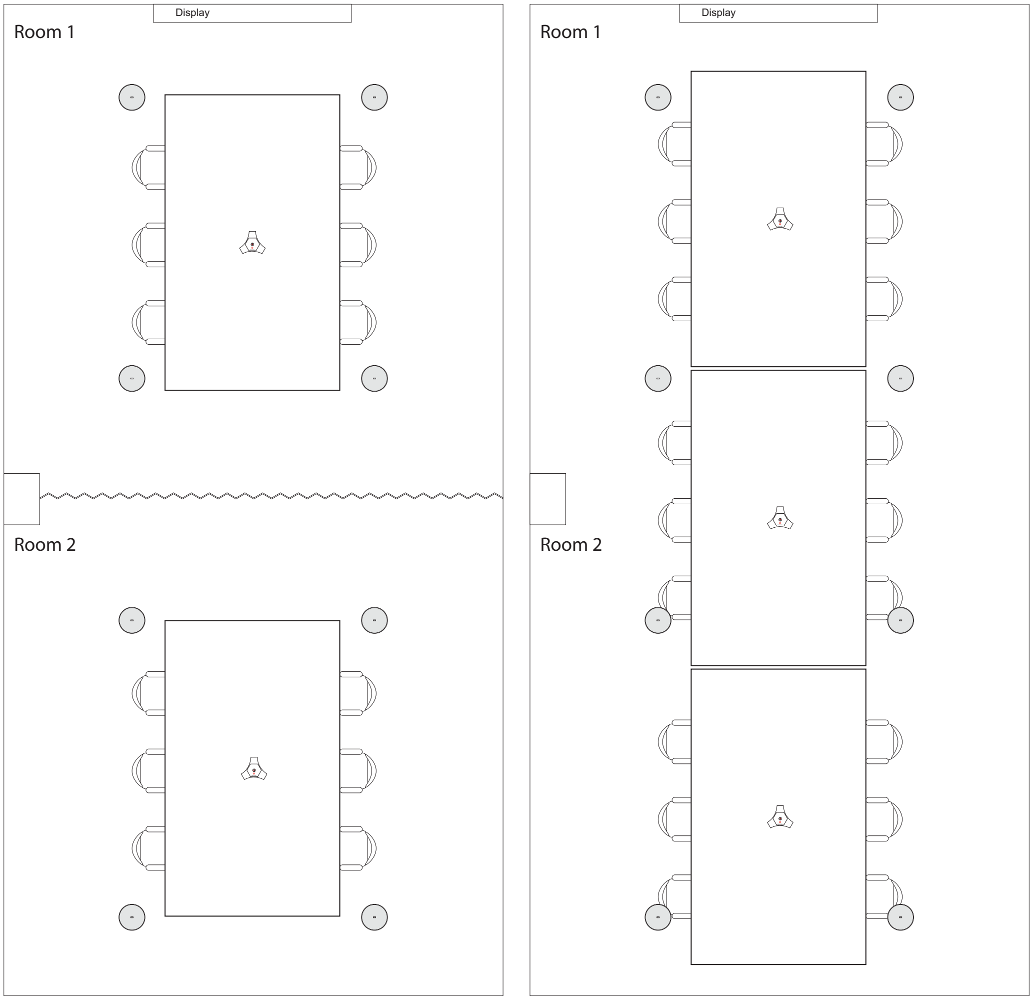
to increase the gain on the faders - making the “VSX8000 In” and “Phone In” channels louder in the local room.

Creating a Room Combining Application Conferencing System

This example shows how to use the SoundStructure products for a room combining application. This example assumes there are two rooms, each with a PSTN line, a program audio feed, a loudspeaker zone, and one digital microphone array in each room. In addition, room 1 also has a Polycom HDX video conferencing system that is used with all microphones when the rooms are combined and only in room 1 when the rooms are split.

The layout for this style of room in the split mode can be seen in the following figure along with the room definitions. When the room is combined, the partition is opened.

Room Layout for Conferencing System



The room configuration will operate as follows.

Combined Mode

In the combined mode, the system is configured as follows:

- All microphones are routed to both telephone lines
- Both telephone lines are routed to the HDX system
- Both telephone lines are routed to the loudspeakers
- Both program audio sources are routed to the loudspeakers
- All microphones are in the same automixer
- The telephones are routed to each other

- There is no reinforcement across zones

Split Mode

In the split mode, the system is configured as:

- Room 1 microphones are in automixer group 1
- Room 1 microphones are routed to the Room 1 telephony transmit and to the HDX codec
- Program audio 1 goes is routed to Room 1 loudspeakers, Room 1 telephony transmit, and the HDX
- Room 1 telephony is routed to Room 1 loudspeakers and to the HDX
- Room 1 HDX remote audio is routed to the Room 1 loudspeakers

Similarly for Room 2:

- Room 2 microphones are in automixer group 2
- Room 2 microphones are routed to the Room 2 telephony transmit
- Program audio 2 is routed to the Room 2 loudspeakers and to the Room 2 telephony transmit
- Room 2 telephony is routed to Room 2 loudspeakers

To create the split and combined settings, there will be two presets called “Split” and “Combine”. These two presets will make it possible to switch easily between the two modes of operation.

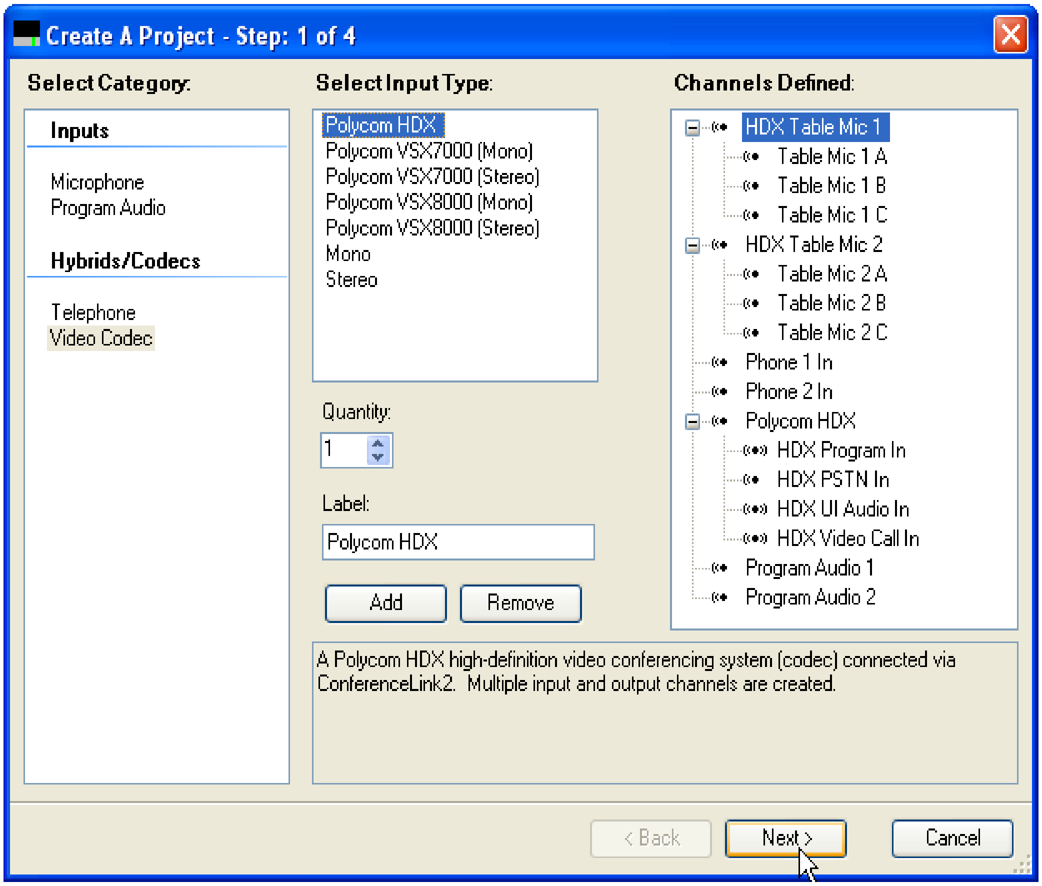
To leverage the control available when using the HDX, this project uses the virtual channel names “Amplifier” and “Mics” (as described in Connecting Over Conference Link2) to allow HDX controllers (such as the HDX IR remote) to be used to mute the microphones in the appropriate zone and adjust volume easily. The “Amplifier” and “Mics” virtual channels will be defined as submixes that can be adjusted with the “Split” and “Combine” presets.

When an HDX video codec is used with SoundStructure, any command to mute the HDX will forward a command to mute the virtual channel “Mics” and if a command is sent to the HDX, the HDX will forward a command to SoundStructure to adjust the fader level on the channel “Amplifier”.

SoundStructure Studio Steps

Step 1 - Select Inputs

The system should be designed in the combined mode with two HDX table microphones, two program audio source, two telephone lines, and a Polycom HDX system.


Create A Project - Step: 1 of 4
Select Category:
Inputs
Microphone
Program Audio
Hybrids/Codecs
Telephone
Video Codec
Select Input Type:
Polycom HDX
Polycom VSX7000 (Mono)
Polycom VSX7000 (Stereo)
Polycom VSX8000 (Mono)
Polycom VSX8000 (Stereo)
Mono
Stereo
Add
Remove
Channels Defined:
HDX Table Mic 1
Table Mic 1 A
Table Mic 1 B
Table Mic 1 C
HDX Table Mic 2
Table Mic 2 A
Table Mic 2 B
Table Mic 2 C
Phone 1 In
Phone 2 In
Polycom HDX
HDX Program In
HDX PSTN In
HDX UI Audio In
HDX Video Call In
Program Audio 1
Program Audio 2
< Back
Next >
Cancel

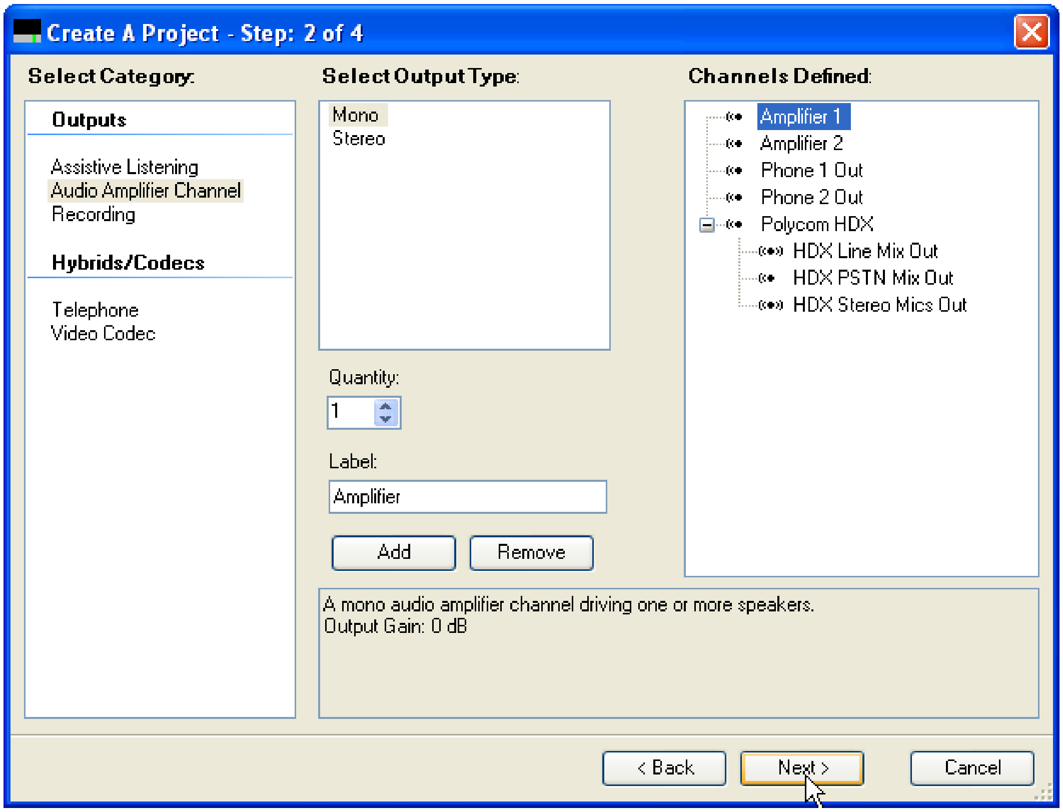
Quantity: 1

Label: Polycom HDX

A Polycom HDX high-definition video conferencing system (codec) connected via ConferenceLink2. Multiple input and output channels are created.

Step 2 - Select Outputs

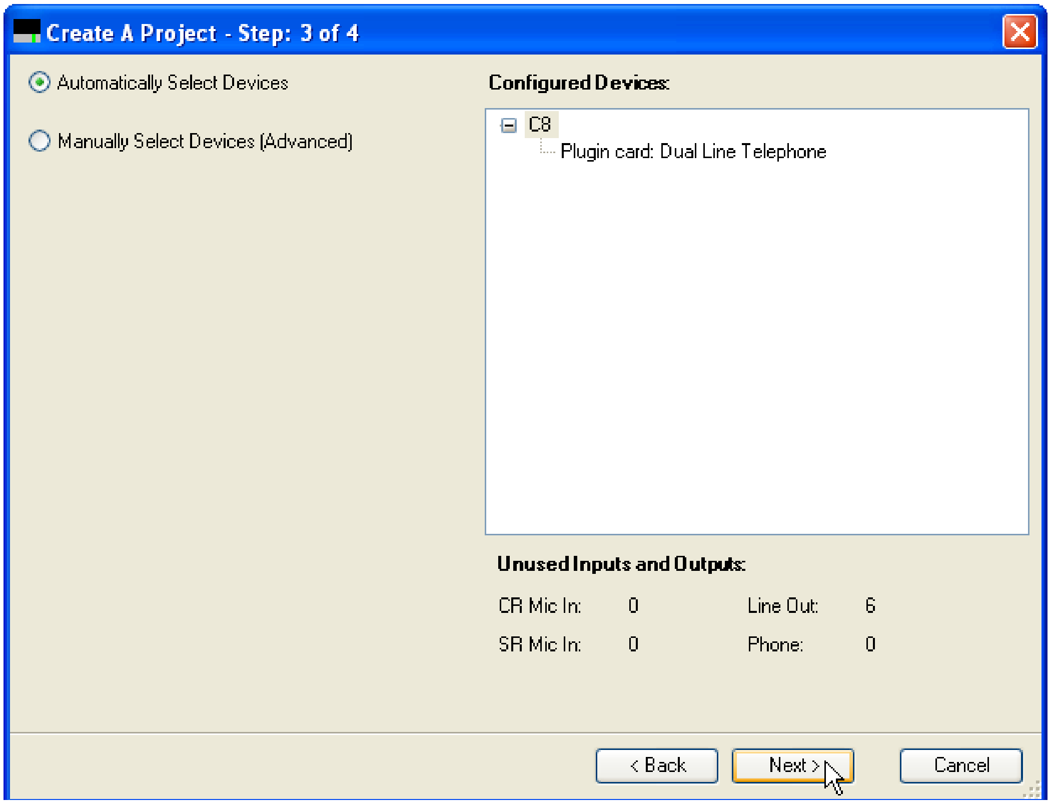
Two mono amplifiers will be selected in this step. The output to the telephone lines and the output to the HDX 9000 were created when their respective input components were added to the system in step 1.


Create A Project - Step: 2 of 4
Select Category:
Outputs
Assistive Listening
Audio Amplifier Channel
Recording
Hybrids/Codecs
Telephone
Video Codec
Select Output Type:
Mono
Stereo
Quantity:
1
Label:
Amplifier
Add
Remove
Channels Defined:
Amplifier 1
Amplifier 2
Phone 1 Out
Phone 2 Out
Polycom HDX
HDX Line Mix Out
HDX PSTN Mix Out
HDX Stereo Mics Out
< Back
Next >
Cancel

A mono audio amplifier channel driving one or more speakers.
Output Gain: 0 dB

Step 3 - Select Equipment

The default equipment selection requires a C8 and a dual telephone line card.



Create A Project - Step: 3 of 4

● Automatically Select Devices

○ Manually Select Devices (Advanced)

Configured Devices:

- C8
  - Plugin card: Dual Line Telephone

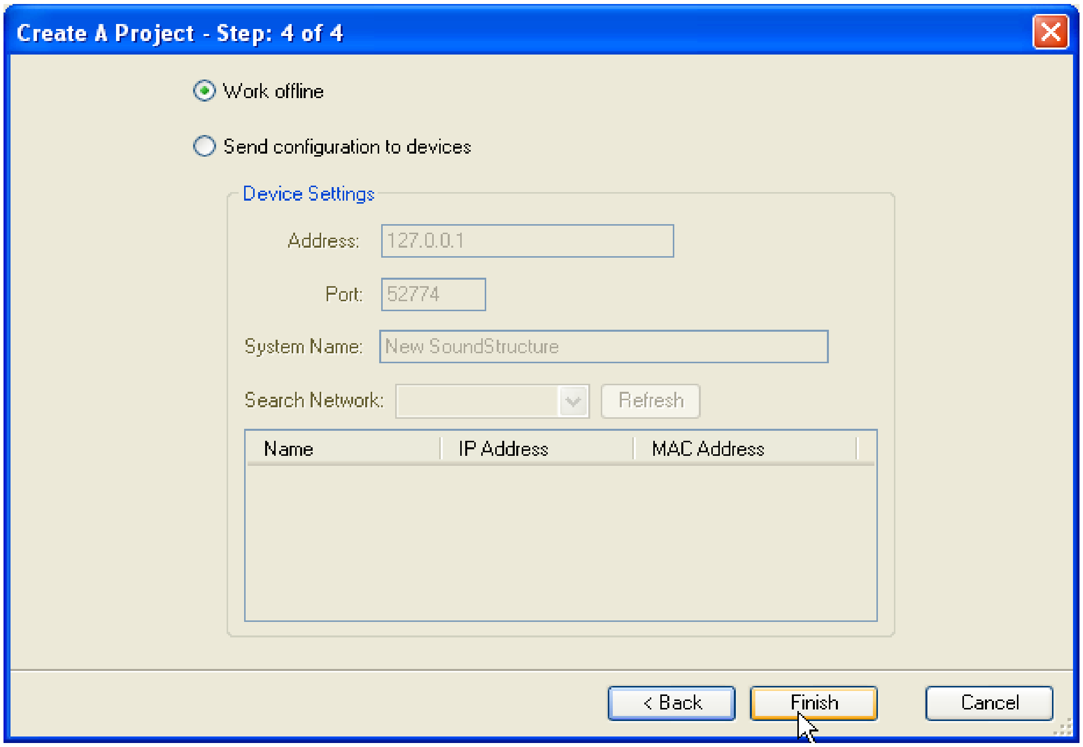
Unused Inputs and Outputs:

CR Mic In:	0	Line Out:	6
SR Mic In:	0	Phone:	0

< Back
Next >
Cancel

Step 4 - Work Offline Or Online

As there are many matrix settings to change, we'll work off line and adjust the crosspoints.



Create A Project - Step: 4 of 4

● Work offline

○ Send configuration to devices

Device Settings

Address: 127.0.0.1

Port: 52774

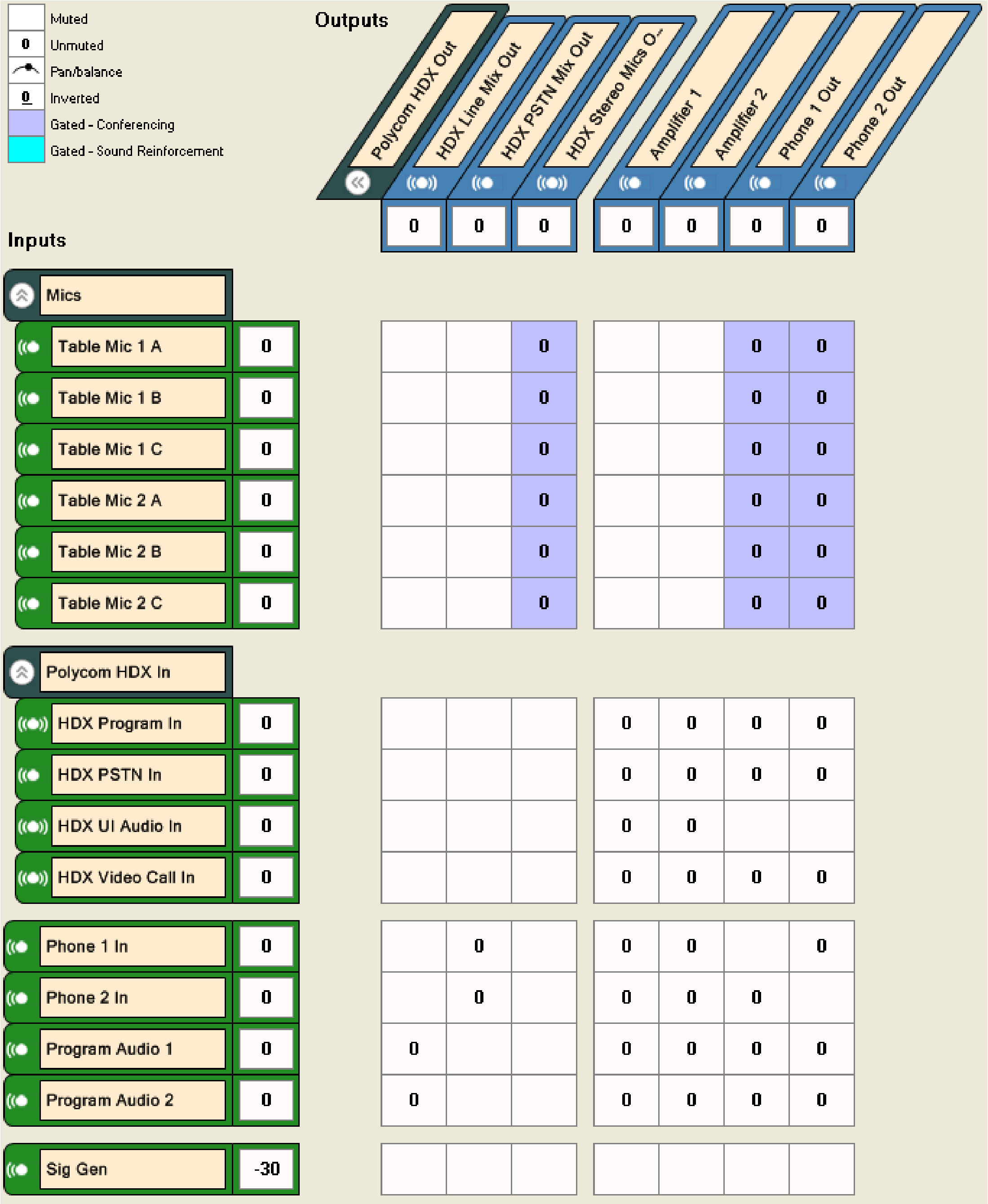
System Name: New SoundStructure

Search Network:

Refresh
Name
IP Address
MAC Address
< Back
Finish
Cancel

Combined Room Settings

The default matrix with the desired inputs and outputs is shown in the following figure.

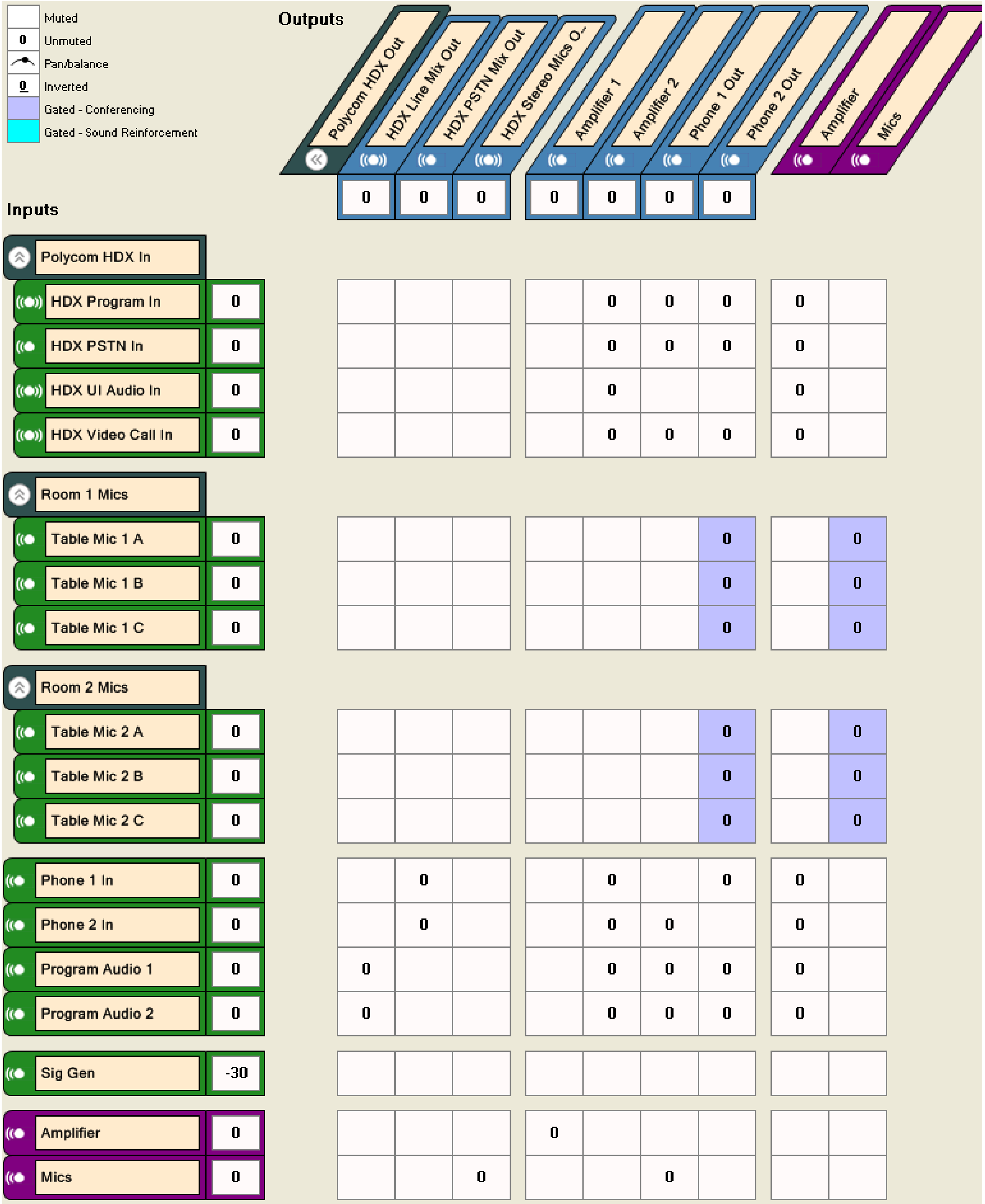

Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Polycom HDX Out
Inputs

Inputs		HDX Line Mix Out	HDX PSTN Mix Out	HDX Stereo Mics O..	Amplifier 1	Amplifier 2	Phone 1 Out	Phone 2 Out
		0	0	0	0	0	0	0
Mics								
Table Mic 1 A	0			0			0	0
Table Mic 1 B	0			0			0	0
Table Mic 1 C	0			0			0	0
Table Mic 2 A	0			0			0	0
Table Mic 2 B	0			0			0	0
Table Mic 2 C	0			0			0	0
Polycom HDX In								
HDX Program In	0				0	0	0	0
HDX PSTN In	0				0	0	0	0
HDX UI Audio In	0				0	0		
HDX Video Call In	0				0	0	0	0
Phone 1 In	0		0		0	0		0
Phone 2 In	0		0		0	0	0	
Program Audio 1	0	0			0	0	0	0
Program Audio 2	0	0			0	0	0	0
Sig Gen	-30							

The next steps are to rename the “Mics” virtual channel to “Room 1 Mics” and change the membership to only include Room 1 microphones, add the group “Room 2 Mics” and add the Room 2 mics to that group.

and create the “Mics” and “Amplifier” submix channels. The updated matrix is shown in the following figure.

Project Matrix Page



☐	Muted
0	Unmuted
⌒	Pan/balance
0	Inverted
🟪	Gated - Conferencing
🟦	Gated - Sound Reinforcement

Outputs

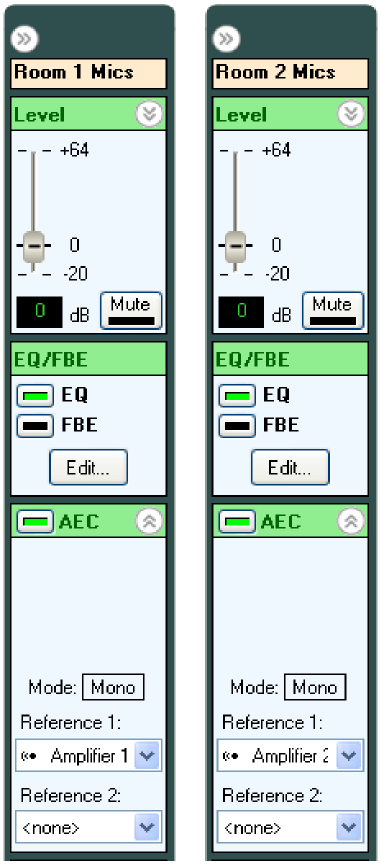
Inputs

		HDX Line Mix Out	HDX PSTN Mix Out	HDX Stereo Mics O...	Amplifier 1	Amplifier 2	Phone 1 Out	Phone 2 Out	Amplifier	Mics
		0	0	0	0	0	0	0		
Polycom HDX In										
HDX Program In	0					0	0	0	0	
HDX PSTN In	0					0	0	0	0	
HDX UI Audio In	0					0			0	
HDX Video Call In	0					0	0	0	0	
Room 1 Mics										
Table Mic 1 A	0							🟪 0		🟪 0
Table Mic 1 B	0							🟪 0		🟪 0
Table Mic 1 C	0							🟪 0		🟪 0
Room 2 Mics										
Table Mic 2 A	0							🟪 0		🟪 0
Table Mic 2 B	0							🟪 0		🟪 0
Table Mic 2 C	0							🟪 0		🟪 0
Phone 1 In	0		0			0		0	0	
Phone 2 In	0		0			0	0		0	
Program Audio 1	0	0				0	0	0	0	
Program Audio 2	0	0				0	0	0	0	
Sig Gen	-30									
Amplifier	0				0					
Mics	0			0			0			

In this matrix, the submix “Amplifier” is used to route the remote audio of the combined system to the “Amplifier 1” virtual channel and the “Mics” submix is used to send the combined microphones to the remote video participants and to the “Phone 1 Out” remote participants. By changing the content of these submixes it is easy to change the Room 1 audio routing.

On the channels page, set the AEC reference for all the Room 1 microphones as “Amplifier 1” and for the room 2 microphones as “Amplifier 2” as shown in the following figure.

AEC Reference for Room 1 Microphones on Channels Page



Room 1 Mics: 0 dB

Mode: Mono

Reference 1: Amplifier 1

Reference 2: <none>

Room 2 Mics: 0 dB

Mode: Mono

Reference 1: Amplifier 2

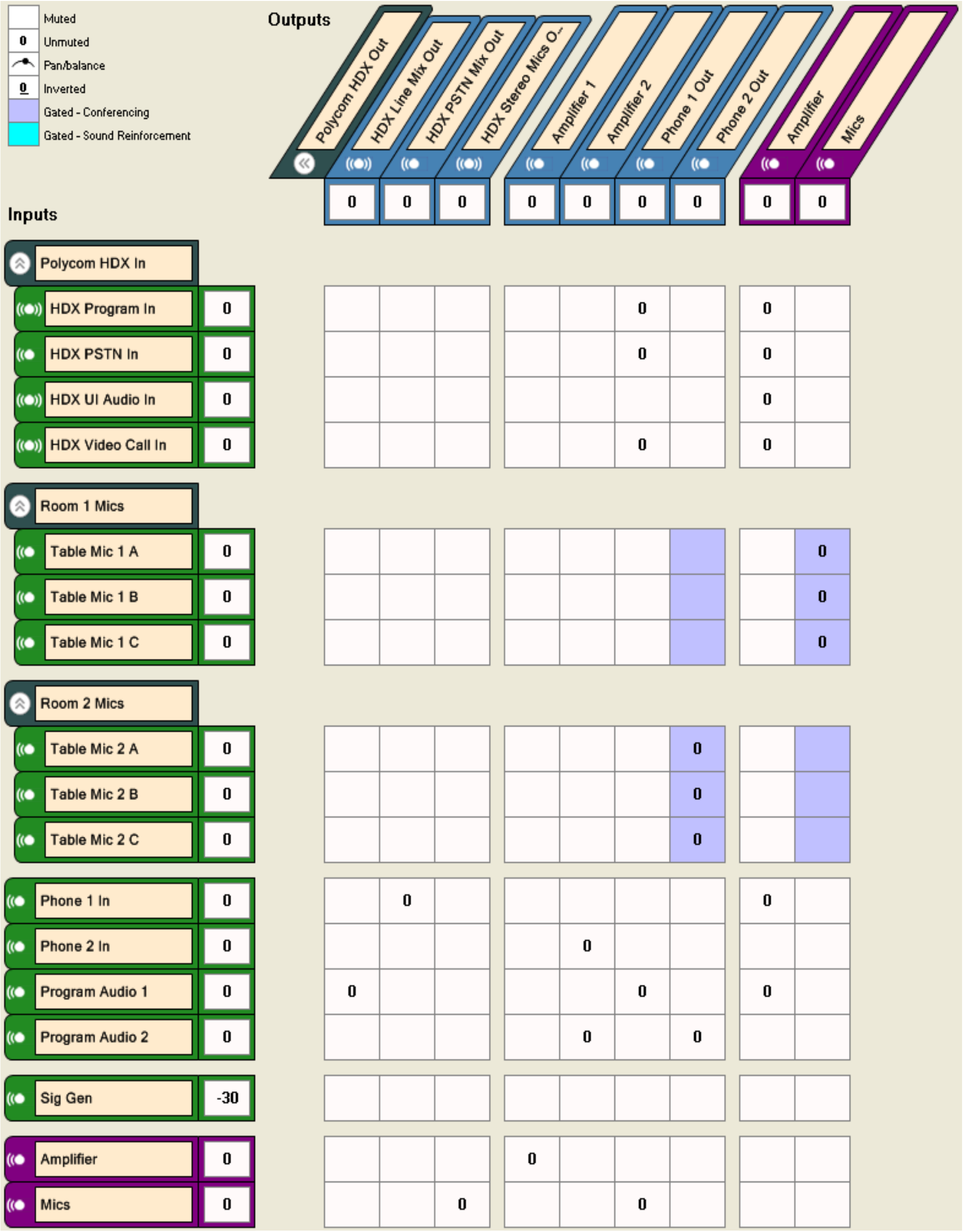
Reference 2: <none>

The routing for Room 2 is done in the matrix without use of the submixes to make it easier to mute or unmute different crosspoints depending on the room combine state. Another approach would have been to create additional submixes for the Room 2 microphones and loudspeaker outputs.

Once the matrix settings are configured, the next step is to save these settings to the “Combine” preset by selecting “Save To New” on the preset page and set the power on preset to be the “Combine” preset.

Split Room Settings

In the split room configuration, the matrix settings must be adjusted to route the audio to meet the original specifications. The following figure shows the routing that keeps the audio from the two rooms completely separate while routing the HDX audio to only Room 1.



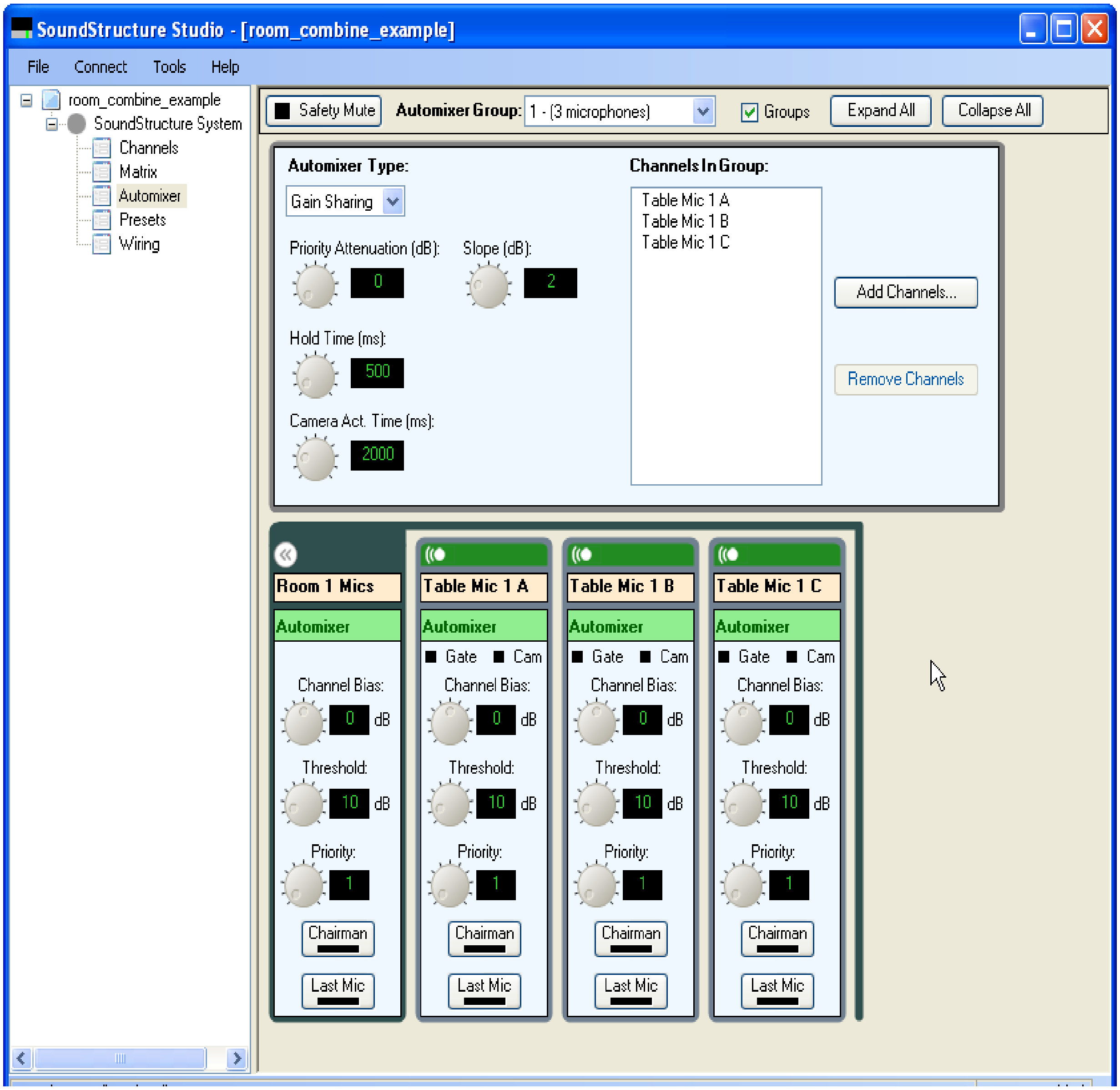
☐	Muted
0	Unmuted
⌒	Pan/balance
0	Inverted
🟪	Gated - Conferencing
🟦	Gated - Sound Reinforcement

Inputs		HDX Line Mix Out	HDX PSTN Mix Out	HDX Stereo Mics O…	Amplifier 1	Amplifier 2	Phone 1 Out	Phone 2 Out	Amplifier	Mics
		0	0	0	0	0	0	0	0	0
Polycom HDX In										
HDX Program In	0						0		0	
HDX PSTN In	0						0		0	
HDX UI Audio In	0								0	
HDX Video Call In	0						0		0	
Room 1 Mics										
Table Mic 1 A	0							🟪		🟪 0
Table Mic 1 B	0							🟪		🟪 0
Table Mic 1 C	0							🟪		🟪 0
Room 2 Mics										
Table Mic 2 A	0							🟪 0		🟪
Table Mic 2 B	0							🟪 0		🟪
Table Mic 2 C	0							🟪 0		🟪
Phone 1 In	0		0						0	
Phone 2 In	0					0				
Program Audio 1	0	0					0		0	
Program Audio 2	0					0		0		
Sig Gen	-30									
Amplifier	0				0					
Mics	0			0			0			

In addition to the matrix settings, the automixer settings must be adjusted to have two automixer groups with the microphones from each room in their respective automixer group.

The automixer settings for the Room 1 mics is shown in the following figure after the Room 2 microphones have been removed.

Automixer Settings for Room 1 Microphones


SoundStructure Studio - [room_combine_example]
File
Connect
Tools
Help
room_combine_example
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Expand All
Collapse All
Add Channels...
Remove Channels
Room 1 Mics
Table Mic 1 A
Table Mic 1 B
Table Mic 1 C
Automixer
Gate
Cam
Chairman
Last Mic

Automixer Group: 1 - (3 microphones)

☑ Groups

Automixer Type: Gain Sharing

Priority Attenuation (dB): 0

Slope (dB): 2

Hold Time (ms): 500

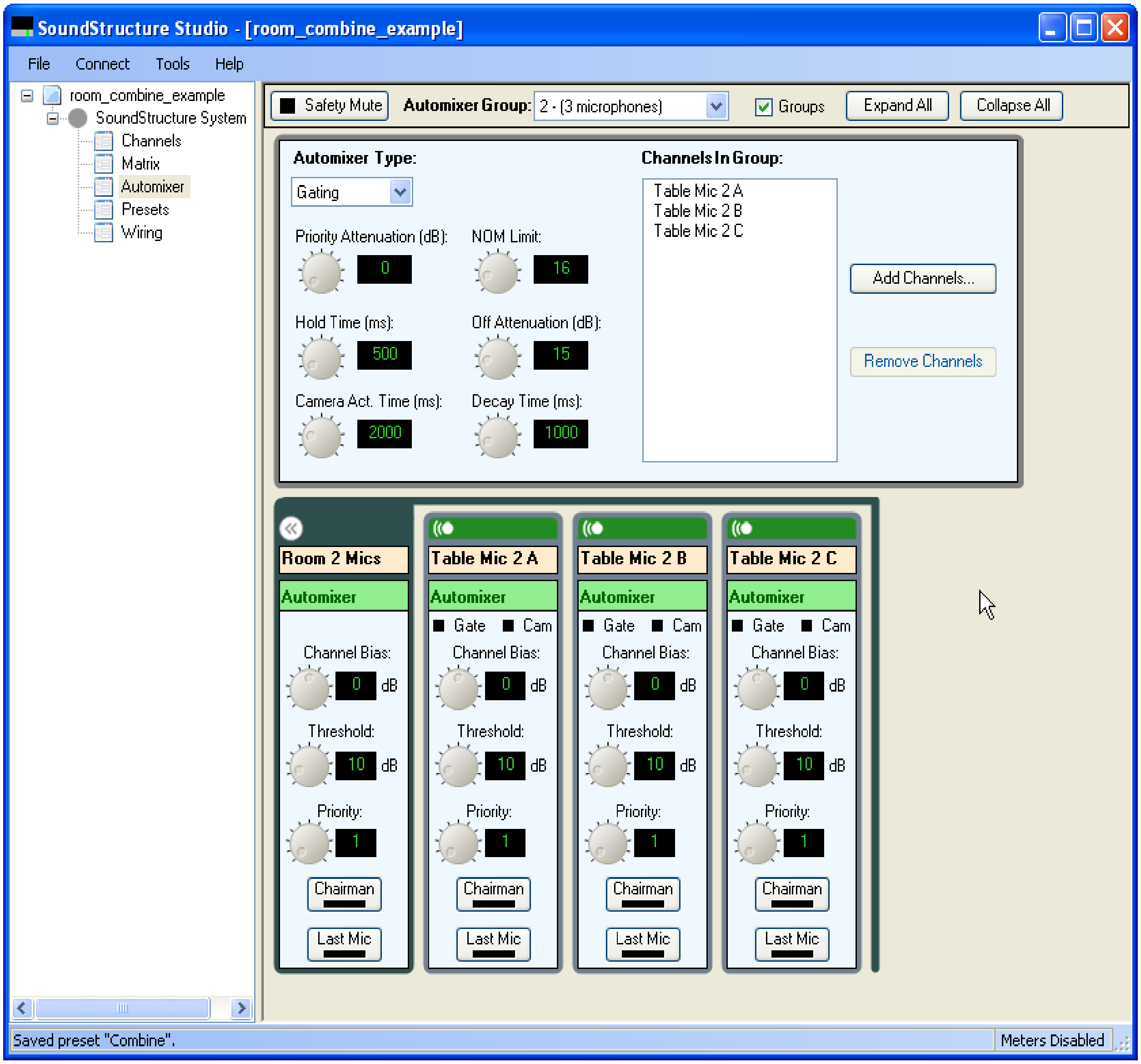
Camera Act. Time (ms): 2000

Channels In Group:
Table Mic 1 A
Table Mic 1 B
Table Mic 1 C

	Room 1 Mics	Table Mic 1 A	Table Mic 1 B	Table Mic 1 C
Channel Bias:	0 dB	0 dB	0 dB	0 dB
Threshold:	10 dB	10 dB	10 dB	10 dB
Priority:	1	1	1	1

The automixer settings for the Room 2 mics is shown in the following figure after setting the Automixer Group to 2 and adding the Room 2 microphones.

Automixer Settings for Room 2 Microphones


SoundStructure Studio - [room_combine_example]
File
Connect
Tools
Help
room_combine_example
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Groups
Expand All
Collapse All
Add Channels...
Remove Channels
Gate
Cam
Chairman
Last Mic

Automixer Group: 2 - (3 microphones)

☑ Groups

Automixer Type: Gating

Priority Attenuation (dB): 0

NOM Limit: 16

Hold Time (ms): 500

Off Attenuation (dB): 15

Camera Act. Time (ms): 2000

Decay Time (ms): 1000

Channels In Group:
Table Mic 2 A
Table Mic 2 B
Table Mic 2 C

	Room 2 Mics	Table Mic 2 A	Table Mic 2 B	Table Mic 2 C
Channel Bias:	0 dB	0 dB	0 dB	0 dB
Threshold:	10 dB	10 dB	10 dB	10 dB
Priority:	1	1	1	1

Saved preset "Combine".

Meters Disabled

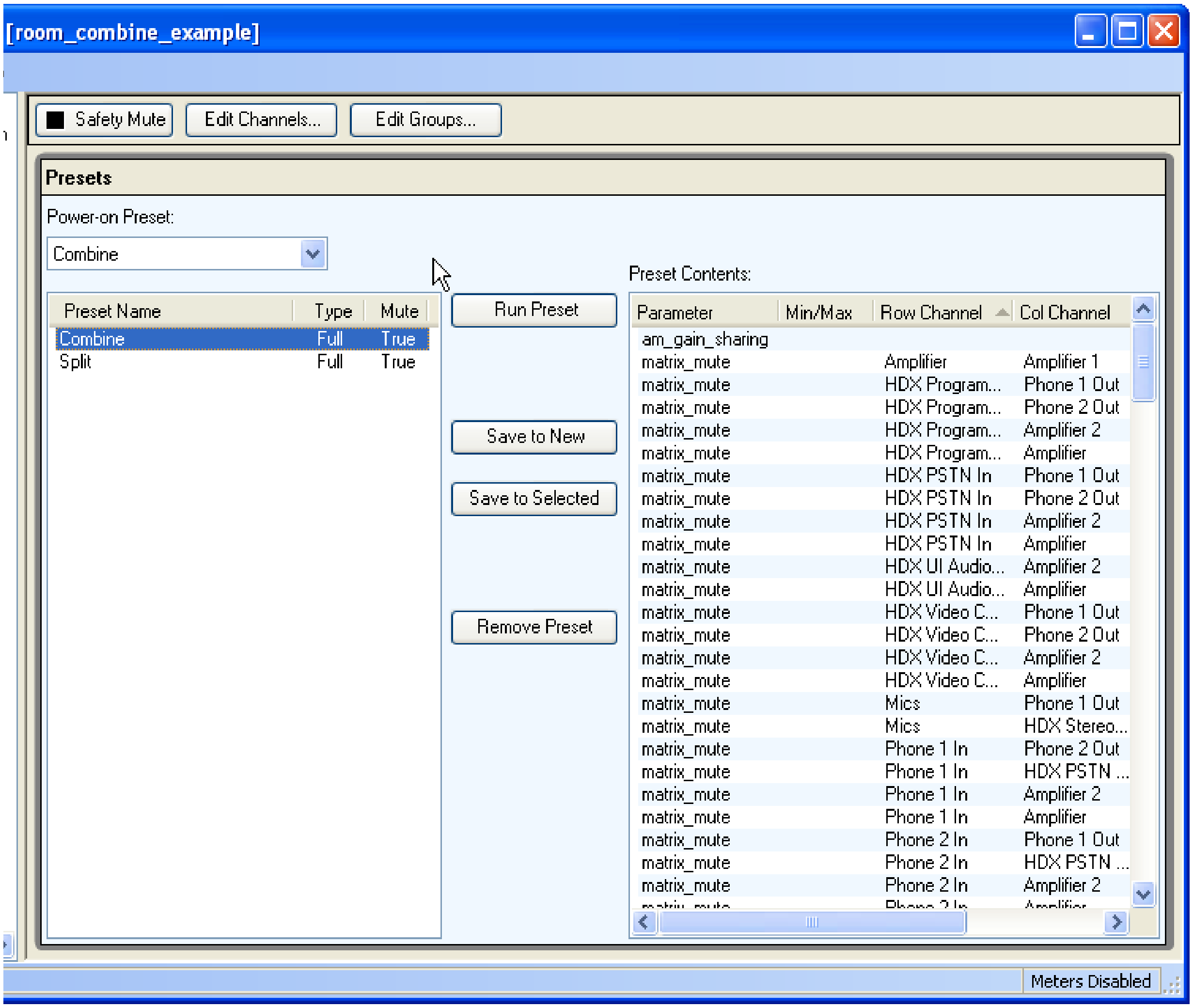
No adjustments need to be made to the echo canceller references because the microphones were configured earlier to use their respective room amplifiers as the AEC reference.

The next step is to save the settings to a new preset and to label that preset “Split”.

Finally, the preset “Power-On” can be removed as those settings do not represent a valid configuration for this design since it contains the settings prior to creating the combined configuration.

Finally, confirm that there is a power on preset - in this example it should be set to be the “Combine” preset as shown in the following figure.

Power on Preset Settings



Power-on Preset: Combine

Preset Name	Type	Mute
Combine	Full	True
Split	Full	True

Preset Contents:

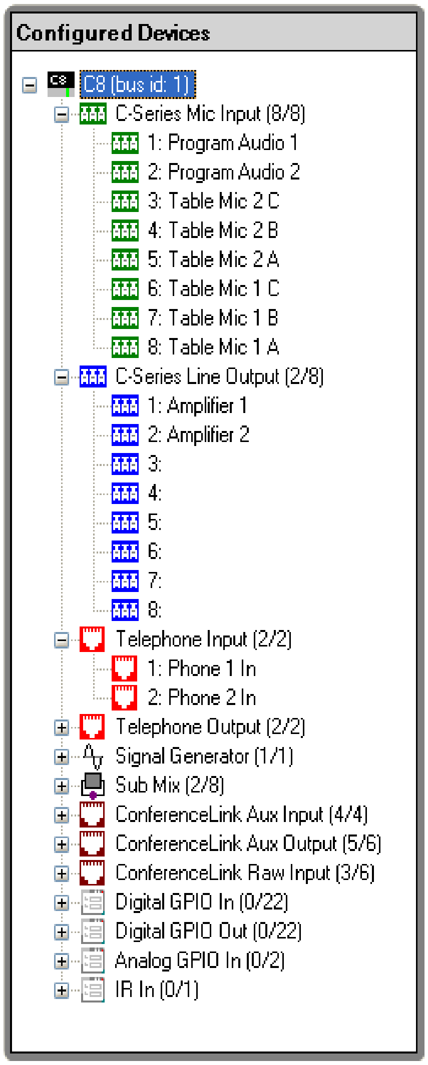
Parameter	Min/Max	Row Channel	Col Channel
am_gain_sharing			
matrix_mute		Amplifier	Amplifier 1
matrix_mute		HDX Program...	Phone 1 Out
matrix_mute		HDX Program...	Phone 2 Out
matrix_mute		HDX Program...	Amplifier 2
matrix_mute		HDX Program...	Amplifier
matrix_mute		HDX PSTN In	Phone 1 Out
matrix_mute		HDX PSTN In	Phone 2 Out
matrix_mute		HDX PSTN In	Amplifier 2
matrix_mute		HDX PSTN In	Amplifier
matrix_mute		HDX UI Audio...	Amplifier 2
matrix_mute		HDX UI Audio...	Amplifier
matrix_mute		HDX Video C...	Phone 1 Out
matrix_mute		HDX Video C...	Phone 2 Out
matrix_mute		HDX Video C...	Amplifier 2
matrix_mute		HDX Video C...	Amplifier
matrix_mute		Mics	Phone 1 Out
matrix_mute		Mics	HDX Stereo...
matrix_mute		Phone 1 In	Phone 2 Out
matrix_mute		Phone 1 In	HDX PSTN ...
matrix_mute		Phone 1 In	Amplifier 2
matrix_mute		Phone 1 In	Amplifier
matrix_mute		Phone 2 In	Phone 1 Out
matrix_mute		Phone 2 In	HDX PSTN ...
matrix_mute		Phone 2 In	Amplifier 2

Wiring Information

The system should be wired according to the information found in the wiring page and shown in the following figure. To wire the system with virtual channels on other physical inputs or outputs, simply drag the channels to their desired locations and then wire the system according to the modified wiring information.

In this example, a single C8 device was used to implement the design. This device is wired as shown in the following figure. The digital microphone arrays use the processing from inputs 3 - 8, leaving inputs 1 and 2 available for the program audio sources. The amplifier outputs for Room 1 and Room 2 are set to outputs 1 and 2 respectively.

Project Wiring Information


Configured Devices

- C8 (bus id: 1)
  - C-Series Mic Input (8/8)
    - 1: Program Audio 1
    - 2: Program Audio 2
    - 3: Table Mic 2 C
    - 4: Table Mic 2 B
    - 5: Table Mic 2 A
    - 6: Table Mic 1 C
    - 7: Table Mic 1 B
    - 8: Table Mic 1 A
  - C-Series Line Output (2/8)
    - 1: Amplifier 1
    - 2: Amplifier 2
    - 3:
    - 4:
    - 5:
    - 6:
    - 7:
    - 8:
  - Telephone Input (2/2)
    - 1: Phone 1 In
    - 2: Phone 2 In
  - Telephone Output (2/2)
  - Signal Generator (1/1)
  - Sub Mix (2/8)
  - ConferenceLink Aux Input (4/4)
  - ConferenceLink Aux Output (5/6)
  - ConferenceLink Raw Input (3/6)
  - Digital GPIO In (0/22)
  - Digital GPIO Out (0/22)
  - Analog GPIO In (0/2)
  - IR In (0/1)

Controlling The System

The system can be controlled in the same manner as the previous examples. The microphones in the combined configuration may be muted and unmuted with the following mute commands.

```
set mute “Mics” 1
set mute “Mics” 0
```

The in-room volume for the remote audio may be increased with the fader command on the phone or video codec audio as follows.

```
inc fader “HDX Video Call In” 1
inc fader “Phone In 1” 1
```

to increase the gain on the faders - making the “HDX Video Call In” and “Phone In 1” channels louder in the local room.

In room volume control of the amplifiers may be accomplished by sending the command

```
inc fader “Amplifier” 1
```

to increment the gain in the combined Amplifier by 1 dB. In the split mode, this command would increment only the Room 1 amplifier by 1dB since only the Room 1 remote audio sources are routed to the “Amplifier” submix in the split mode.

Troubleshooting

This chapter presents a series of situations and troubleshooting steps to resolve the situation. Troubleshooting is most effective when problems can be isolated, reproduced, and then resolved one at a time. This “divide-and-conquer” approach will be used in this chapter.

Audio Troubleshooting

Many audio problems can be traced to the following issues:

1. Wiring issues - the system is wired differently from how SoundStructure Studio thinks the system should be wired.
2. Audio isn’t routed properly through the matrix to the desired outputs
3. The signal is muted at inputs or outputs, or possibly safety mute is enabled
4. The gain structure for the signal is not appropriate - too much or too little gain is applied at the inputs or outputs or the input or output fader has a value significantly different from 0 dB. See Installing SoundStructure Devices for guidelines on setting the input and output gains
5. The gain on the amplifier that drives audio into the local room is not configured properly. The amplifier level should be adjusted after the remote audio input levels have been adjusted on the SoundStructure.
6. Physical wiring issues - phoenix connectors are not terminated properly or inputs are plugged into outputs and outputs are plugged into inputs by mistake - remember the inputs are on the bottom row of phoenix connectors and the outputs are on the top row of phoenix connectors.

In most cases, simplifying the system, for instance by muting all but one microphone, can be used to isolate a particular issue.

Below are some common issues with associated steps for resolving the issue.

Local participants Can’t Hear Remote Participants

Check that the audio from the remote participants is routed through the matrix to the local amplifier outputs.

Is the amplifier turned on? Can other sources of audio be heard in the local room? Add a Signal Generator from the Edit Channels control and route the signal generator to the amplifier virtual channel.

Check that the wiring for the amplifier virtual channel on the wiring page matches how the system is actually wired.

Check that the audio from the remote participants is not muted either locally or at the remote site.

Remote Participants Can’t Hear Local Participants

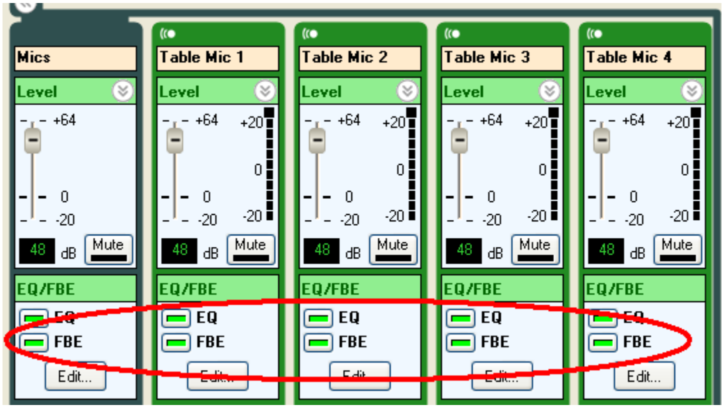
Check that the audio from the local participants is routed through the matrix to the remote participants.

Are the microphones unmuted? Can microphones be routed to the amplifier (lower the gain at the crosspoint!) and the microphones heard in the local room?

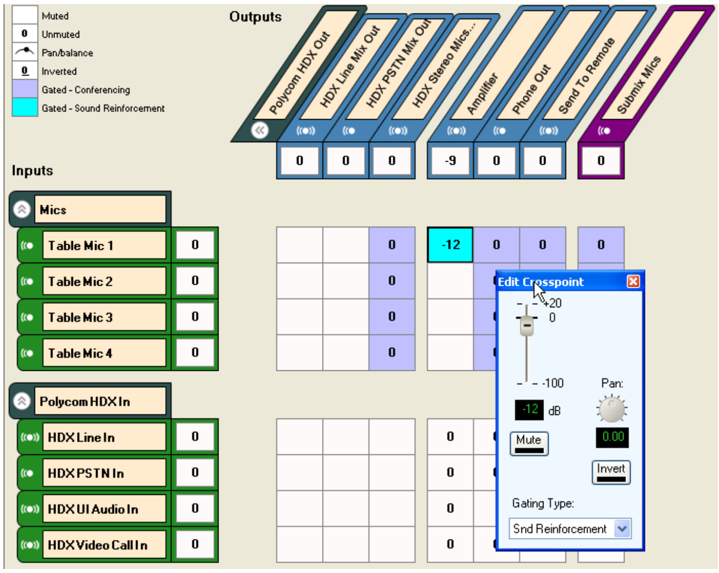
Check that the wiring for the remote virtual channels on the wiring page matches how the system is actually wired.

Feedback Elimination Doesn’t Seem To Be Operational - Feedback Can Be Heard Locally

Ensure the feedback eliminator is enabled on the microphones being used for reinforcement as shown in the following figure.


Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Level
+64
+20
0
-20
48
dB
Mute
EQ/FBE
EQ
FBE
Edit...

Also ensure the sound reinforcement signal path is selected at the matrix crosspoint. There should be a light blue background on the crosspoints routing the microphones to be reinforced to the audio amplifier as shown in the following figure where “Table Mic 1” is routed to the “Amplifier” virtual channel.


Muted
Unmuted
Pan/balance
Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Polycom HDX Out
HDX Line Mix Out
HDX PSTN Mix Out
HDX Stereo Mics
Amplifier
Phone Out
Send To Remote
Submix Mics
Inputs
Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Polycom HDX In
HDX Line In
HDX PSTN In
HDX UI Audio In
HDX Video Call In
-9
-12
Edit Crosspoint
+20
0
-100
Pan:
dB
0.00
Mute
Invert
Gating Type:
Snd Reinforcement

How Do I Enable Auto Gain Control Or Noise Cancellation On The Program Audio Material And Video Codec Audio?

Customizing SoundStructure Designs, in the Processing Noise Cancellation section, describes how to select the Line Input “ungated” type, and then how to use that signal processing path in the matrix. Once the processing path is selected, the AGC and noise cancellation can be independently enabled for those channels.

How To Set The Audio Amplifier Default Level?

Installing SoundStructure Devices describes the steps to take to ensure a good level to the audio amplifier. This involves setting the proper output level from the SoundStructure device and adjusting the volume of the amplifier until a good level is heard in the room. Volume adjustments can then be made with the output fader on the channels connected to the audio amplifier.

How Do I Get More Than One Signal Generator?

SoundStructure devices only support one signal generator per device. If more than one Signal Generator is required, multiple devices must be linked over OBAM and the signal generators on each of those devices may be used independently.

Echo Troubleshooting

Many echo problems can be traced to:

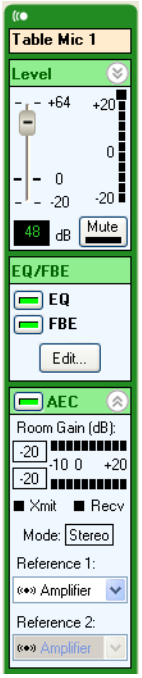
1. Check loop-back echo. A matrix cross-point may have been inadvertently unmuted, causing a direct replica of the audio to be heard remotely.
2. AEC Reference is setup incorrectly (see Customizing SoundStructure Designs). Note: AEC reference needs to include ALL the remote audio sources. Any remote audio that is not part of the reference will hear echo going back to that site.
3. Room gain is too high (see Installing SoundStructure Devices). A typical method to reduce the room gain is to provide a better input level to the SoundStructure device and lower the amplifier level. Others may require a different placement of loudspeakers and microphones.
4. Audio has too much non-linear distortion. If the playback audio is clipping the loudspeaker, the resulting echo picked up at the microphone can also become nonlinearly distorted. In this case, the AEC will not adapt to the room echo correctly. One way to resolve this is to lower the amplifier level or the digital gain inside the SoundStructure of the audio path going to the amplifier output.

The Remote People Hear Echo Of Their Voices From The Local Room

Mute the local microphones and ensure the echo is removed for the remote participants when the local microphones are muted. Unmute the local microphones and ensure the echo has returned.

If the echo is present when the microphones are unmuted and not there when the local microphones are muted, it is likely an acoustic echo canceller configuration issue with the local room. If the echo is still there when the microphones are muted, it is not an acoustic echo issue and may be an issue with wiring or with routing through the matrix.

Mute all the microphones except for one and on the unmuted microphone, check the value of the AEC reference. In the following figure the AEC reference is set to the “Amplifier” stereo virtual channel.



Mode: Stereo

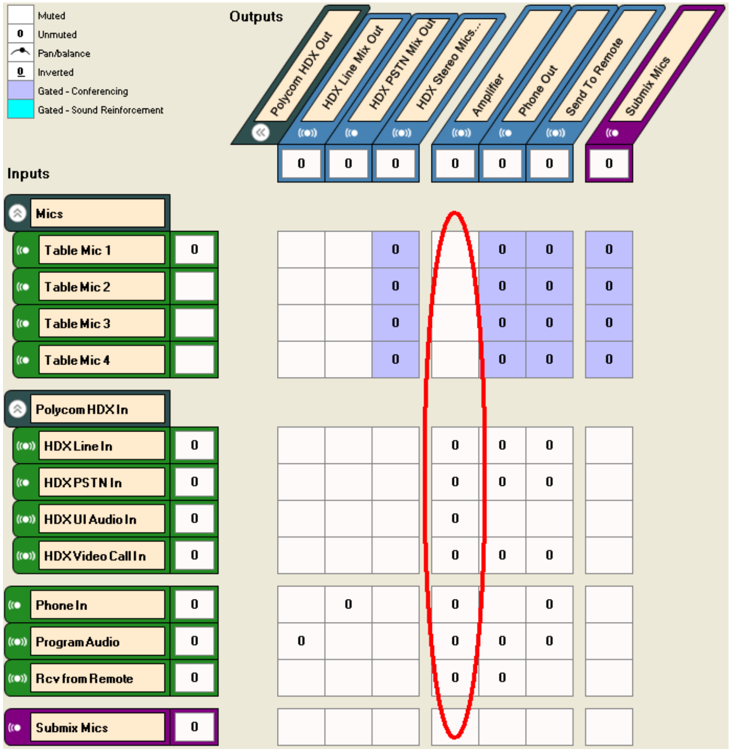
Reference 1: Amplifier

Reference 2: Amplifier

Next, check the matrix to ensure the “Amplifier” virtual channel includes the remote audio sources. An example of the “Amplifier” channel and all the remote audio sources that make up the “Amplifier” channel is shown in the following figure. Notice that the audio from the Polycom Video Codec, the telco audio, the program audio, and the audio from the remaining remote source are all part of the “Amplifier” virtual channel and consequently used as the AEC reference.

If the AEC reference does not include a particular remote audio source, then whenever that remote audio source is active, there will be residual echo sent back to that remote source. For example, if the telephone signal is not part of the reference, then when the telephone participants speak, they will hear an echo of their voice being sent back to themselves.

If the reference is set properly, and the reference is configured properly in the matrix, the next step is to check the room gain of the system and make sure it is not too high. Installing SoundStructure Devices discusses acceptable room gain levels, and how to reduce room gain by lowering the audio amplifier level and increasing the input gain on the remote audio coming into the SoundStructure to ensure the signal levels are at a reasonable level.

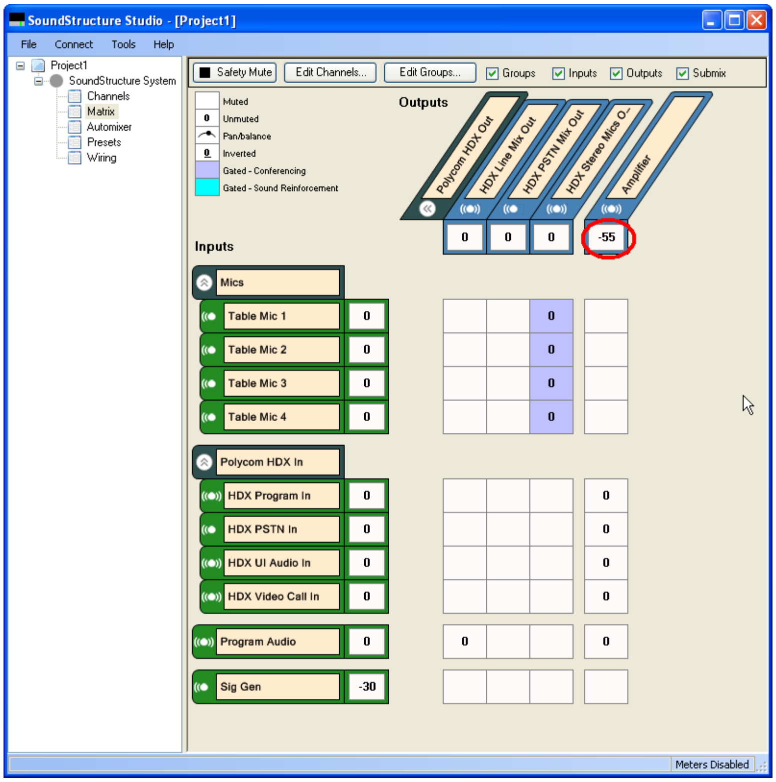


☐ Muted
0 Unmuted
Pan/balance
0 Inverted
Gated - Conferencing
Gated - Sound Reinforcement

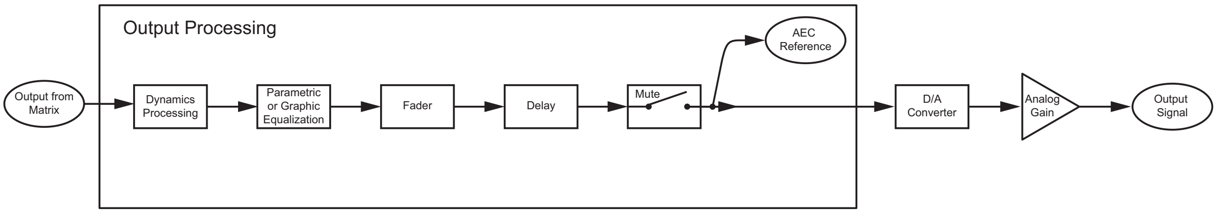
Inputs	Fader	HDX Line Mix Out	HDX PSTN Mix Out	HDX Stereo Mics...	Amplifier	Phone Out	Send To Remote	Submix Mics
Outputs		0	0	0	0	0	0	0
Mics								
Table Mic 1	0			0		0	0	0
Table Mic 2				0		0	0	0
Table Mic 3				0		0	0	0
Table Mic 4				0		0	0	0
Polycom HDX In								
HDX Line In	0				0	0	0	
HDX PSTN In	0				0	0	0	
HDX UI Audio In	0				0			
HDX Video Call In	0				0	0	0	
Phone In	0		0		0		0	
Program Audio	0	0			0	0	0	
Rcv from Remote	0				0	0		
Submix Mics	0							

If the reference is set properly and includes all the remote audio sources and there is still an echo heard by the remote participants, the next step is to understand how the amplifier output fader is set.

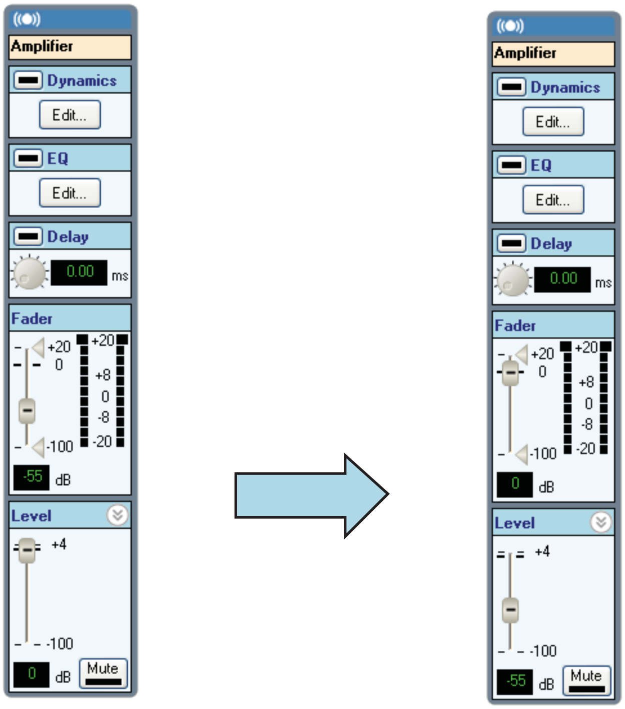
In some applications, the line level outputs of the SoundStructure could be connected to the inputs of a microphone-only device that requires the outputs of the SoundStructure to be attenuated significantly to be compatible with the microphone level inputs. If the output fader on the amplifier channel is used to attenuate the amplifier signal as shown in the figure below and the AEC reference is also set to the amplifier output, then the AEC reference would also be attenuated by the fader amount.


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
SoundStructure System
Channels
Matrix
Automixer
Presets
Wiring
Safety Mute
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Muted
Unmuted
Pan/balance
Inverted
Gated - Conferencing
Gated - Sound Reinforcement
Outputs
Polycom HDX Out
HDX Line Mix Out
HDX PSTN Mix Out
HDX Stereo Mics O...
Amplifier
0
0
0
-55
Inputs
Mics
Table Mic 1
Table Mic 2
Table Mic 3
Table Mic 4
Polycom HDX In
HDX Program In
HDX PSTN In
HDX UI Audio In
HDX Video Call In
Program Audio
Sig Gen
-30
Meters Disabled

Because the AEC reference is available after the fader as presented in Introducing SoundStructure Design Concepts and shown in the following figure, the result is that the AEC reference is also attenuated and therefore the echo canceller would not be able to remove the echo because the reference level is attenuated too much.


Output Processing
AEC Reference
Output from Matrix
Dynamics Processing
Parametric or Graphic Equalization
Fader
Delay
Mute
D/A Converter
Analog Gain
Output Signal

The solution to this issue is to use the line output gain instead of the fader to attenuate the signal to match it to the signal level requirements of the next piece of equipment in the signal chain. Changing these settings are shown in the following figure. The result of this is that the proper signal levels are presented to the echo canceller and the output signal levels are attenuated appropriately.



The Local People Hear Echo Of Their Voices From The Remote Room

This problem is most likely with the remote room's acoustic echo canceller. Have the remote participants mute their microphone to see if the echo is removed, if so, troubleshoot the remote room's AEC by following the instructions given previously. For remote people follow the instructions for *The remote people hear echo of their voices from the local room* issue described above.

If muting the remote participants microphones did not remove the acoustic echo issue, then check the routing of the remote audio matrix to ensure the audio from the local room to the remote room is not being sent directly back to the local room.

Room Gain Is High - What Does It Mean?

Installing SoundStructure Devices discusses room gain and what the acceptable and expected levels should be for ceiling microphones vs. table microphones. Room gain above +10 dB should be reviewed to ensure the input gain on the remote audio sources is high enough to get the remote sources to the 0 dBu nominal signal level expected by the SoundStructure devices.

API Troubleshooting

When using TeraTerm 3.1 and connecting over Telnet, why do I have to select CR-LF termination for commands sent to SoundStructure and not just CR termination?

As described in Appendix A, SoundStructure devices accept commands sent to it with either CR or CR-LF terminations. What we noticed is that when using Tera Term in telnet mode, Tera Term terminates commands transmitted to SoundStructure with two bytes - CR and a Null character - even though only the CR termination is selected in the Tera Term user interface. This is a bug within Tera Term. The result is that all commands that are sent to SoundStructure start with the Null character which will not be interpreted as a valid command.

To resolve this issue, select the CR-LF transmission termination option within the Tera Term user interface when using telnet connections. When using Tera Term in serial mode, either CR or CR-LF line terminations operate properly.

What Does The Error “invalid action specified” Message Mean?

Typical actions for a command include the values of set, get, inc, dec, tog to respectively set, get, increment, decrement, or toggle the value of a parameter. If the action is not typed properly or is not in lower case, this error message may occur. Correct the syntax or case and try again.

What Does The Error “device ID not specified” Message Mean?

For commands that require a device ID to be specified, not including the device ID will cause this error message. As an example, sending the command:

```
get ser_baud
```

will generate this error message. The proper syntax for this command is

```
get ser_baud 1
```

where 1 is the device ID of the SoundStructure system.

To resolve this issue, adjust the syntax of the command to include the device ID.

What Does The Error “virtual channel or virtual channel group label not quoted” Message Mean?

When a virtual channel name is used in a command, it must be surrounded with double quotes. If the virtual channel name or virtual channel group name is not in double quotes, then this error message will occur. For example, the command

```
set mute Table Mic 1 1
```

will cause this error message. Fix this syntax by putting double quotes around the virtual channel name such as with the command

```
set mute “Table Mic 1” 1
```

and the system will work properly.

What Does The Error “no virtual channel or virtual channel group with that label exists” Message Mean?

If an API command references a virtual channel name that doesn’t exist then this message will be received. Correct the spelling of the virtual channel name, or create the virtual channel if it doesn’t exist, and try again.

What Does The Error “invalid parameter name” Message Mean?

If the API command sent to the SoundStructure device is not correct, perhaps due to a typo on the command or the improper syntax used, the SoundStructure device will return with an error 38.

What Does The Error “parameter argument not specified” Mean?

If the command syntax of the command is not followed such as specifying too many parameters or not enough parameters, this error message may occur. As an example, setting the baud rate of a SoundStructure device requires specifying the device ID as in the following example.

```
set ser_baud 1 9600
```

If the device ID is not specified, such as with the following example:

```
set ser_baud 9600
```

then this error message will occur.

What Does The Error “invalid parameter argument” Message Mean?

If the argument for the command is not correct, for instance trying to set the mute state of microphone to the value 3 when the only valid values are 0 or 1, then this error message may occur.

Why Won’t The Control System Mute The Microphones?

Check that the command from the control system isn’t generating one of the error message described above. Next ensure that the control system is connected to the SoundStructure device over RS-232 or Ethernet and able to send commands to the SoundStructure device.

If muting the microphones by using the default virtual channel group “Mics”, the syntax of the command should be:

```
set mute “Mics” 1
```

and

```
set mute “Mics” 0
```

to mute and unmute, respectively the microphones. This command should generate a series of command status messages that report the mute state of the individual virtual channels that are in the virtual channel group as well as an overall status of the virtual channel groups mute status.

Remember that the API must be in lower case and that the virtual channel names are case sensitive.

I Muted All The Members Of My Virtual Channel Group, Why Don’t I Get A Group Acknowledgment That The Virtual Channel Group Is Muted?

The way that virtual channels and virtual channel groups work is that when a virtual channel group is muted or the gain adjusted, for example, all the channels in the group get are set to the same value and all the virtual channels in the group reply with command acknowledgments reflecting their new value. If the members of the group are set to the same value, there is no command acknowledgment that comes from the group. The only way to get a group acknowledgment is to send a command to the group.

Where Do I Get More Info About The API?

Appendix A in this manual describes the command API syntax and the file soundstructure-parameters.html on the CD-ROM includes the full list of parameters that can be adjusted for the SoundStructure devices. The full API can be found also by pointing your web browser at the IP address of the SoundStructure device.

Do Commands Need To Be In Upper Or Lower Case?

All API commands must be in lower case. Sending upper case commands will cause error messages to be returned by the SoundStructure device.

Virtual channel and virtual channel group names can be in mixed case. Remember that virtual channel names are case-sensitive - “Table Mic 1” and “table mic 1” are two different virtual channel names.

I’ve Tried Everything And I Still Can’t Connect To The SoundStructure Device

Reboot the SoundStructure device and see if it is possible to connect to the device either via RS-232 or Ethernet. If so, check the Polycom website for a newer version of firmware and release notes to see what issues may be been resolved.

RS-232 Troubleshooting

I Can’t Connect Over RS-232 To The System, How Do I Connect?

Check that the baud rate between the PC or Control system and the SoundStructure device are set to the same value. Baud rates above 9,600 baud should have hardware flow control enabled on both the SoundStructure device and the control system or local PC.

How Do I Set The Baud Rate? What If I Can’t Connect Over RS-232?

By default the baud rate of the SoundStructure devices is set to 9600 bps. Try connecting the device at this baud rate over the serial port.

There is an API command ser_baud that can be used to set the baud rate of the SoundStructure device. To adjust baud rate, send the command

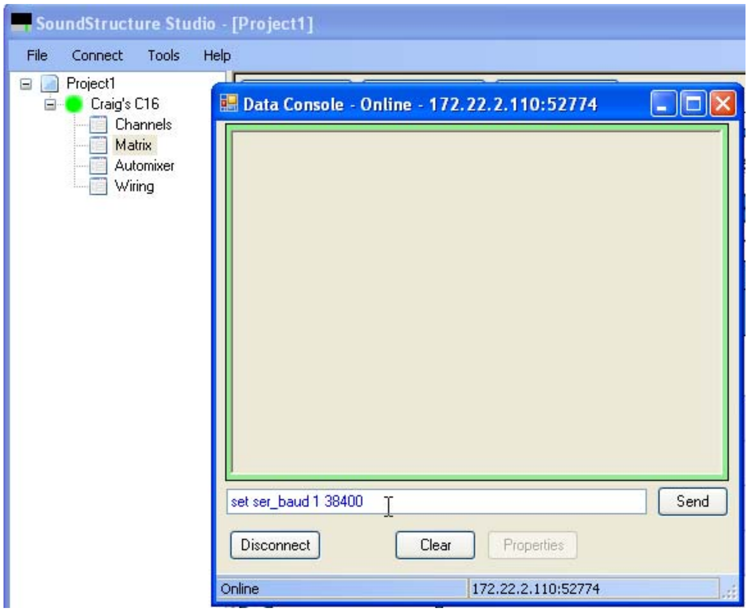
```
set ser_baud 1 9600
```

where 1 is the device ID of the device.

Remember if you change the baud rate and are connected over RS-232 at the previous baud rate, you will have to change the baud rate on your PC terminal program or Control System to continue talking to the device over the RS-232 interface.

The RS-232 cable requires straight through wiring as shown in Creating Advanced Applications.

The baud rate may be set using either the Ethernet interface or RS-232 interface. Connect to the device as described in Installing SoundStructure Devices and open the console window by right clicking on the device name in SoundStructure Studio and type the command directly into the console window as shown in the following figure.


SoundStructure Studio - [Project1]
File
Connect
Tools
Help
Project1
Craig's C16
Channels
Matrix
Automixer
Wiring
Data Console - Online - 172.22.2.110:52774
Send
Disconnect
Clear
Properties
Online
172.22.2.110:52774

```
set ser_baud 1 38400
```

What Is Flow Control And How Does It Work?

Hardware flow control on the SoundStructure device requires two additional handshaking signals, CTS and RTS, in the RS-232 cable to ensure data is received before additional data is sent. This prevents the serial port from dropping data due to not being ready for new data. Flow control literally controls the flow of data between two serial devices.

If hardware flow control is used - and it is recommended that you use flow control on data rates above 9600 baud - then it should be enabled on both the SoundStructure device and the device that is connected to the SoundStructure device.

Hardware flow control may be enabled on a SoundStructure device with the API command

```
set ser_flow 1 hw
```

and may be removed with the command

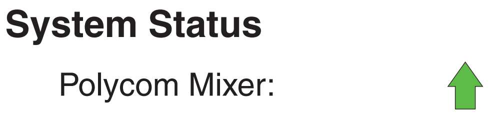
```
set ser_flow 1 none
```

where 1 is the device ID.

Polycom Video Codec Integration

How Do I Know the Polycom Video Codec System Is Connected Properly to SoundStructure?

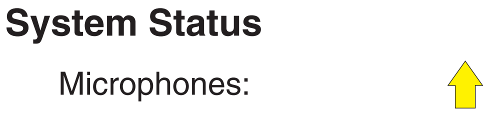
The Polycom Video Codec Diagnostics and System Status screen shows whether the SoundStructure device (labeled as Polycom Mixer) has been detected. A green arrow associated with the Polycom Mixer, as shown in the following figure, indicates the systems have detected each other and are working properly.



System Status

Polycom Mixer:

Alternatively if the message appears as below, then the two systems have not detected each other and are not communicating over Conference Link2.



System Status

Microphones:

If the SoundStructure system hasn't been detected by the Polycom Video Codec, then remove the CLink2 cable from the rear of the SoundStructure device and reconnect it. Refresh the Video Codec UI page by moving from the page and then back to the page.

The connection status can also be viewed within the System Information page on the Polycom Video Codec. If the status shows Polycom Mixer then the system has connected properly to the SoundStructure.

How Do I Connect Multiple Polycom Video Codecs to SoundStructure?

The CLink2 integration only supports the digital integration of one Polycom Video Codec system connected to a SoundStructure device. To connect additional Video Codec systems, they must use analog cables to connect physical inputs and outputs of the SoundStructure device to the Polycom Video Codec systems. Within SoundStructure Studio select multiple VSX8000 systems (mono or stereo depending on your application) to create the default inputs and outputs to integrate via analog signals to the Polycom Video Codecs.

If I Change Volume On SoundStructure, Why Don't I See The Video Codec Volume Bar Update?

As described in Connecting Over Conference Link2, volume commands from the Polycom Video Codec send commands to the SoundStructure device and adjust the fader on the "Amplifier" virtual channel within the SoundStructure system. If the fader control on the "Amplifier" channel is adjusted independently on the SoundStructure system, a command is not sent to the Polycom Video Codec and consequently the Polycom Video Codec will not update the volume bar on the screen.

If using a control system to adjust volume in a system that includes both a Video Codec and a SoundStructure, have the control system adjust the volume on the Video Codec system and the SoundStructure fader control for the virtual channel "Amplifier" will track to that value.

If I Mute On The SoundStructure, Why Doesn’t The Mute Icon Appear On The Video Codec?

As described in Connecting Over Conference Link2, mute commands from the Polycom Video Codec send commands to the SoundStructure device and adjust mute status of the “Mics” virtual channel group within the SoundStructure system. If the mute status of the “Mics” group is adjusted independently on the SoundStructure system, a command is not sent to the Polycom Video Codec and consequently the Polycom Video Codec will not update the mute status on the screen.

If using a control system to change the local global mute status in the system, have the control system adjust the mute state on the Video Codec system and the SoundStructure mute state for the “Mics” group will track to that state.

Telco Troubleshooting

Phone Won’t Go Off Hook Or I Don’t Hear Dial Tone

Check that the phone line from the PBX or central office is plugged into the LINE port on the rear of the SoundStructure device.

Use SoundStructure Studio and from the Channels Page select the phone *Settings...* button to open a telephone keypad. Click the handset icon to take the phone off hook.

Check that the virtual channel name used for the telephone channel matches the name used within SoundStructure Studio to create the telephone channel.

Check that you are able to control other aspects of the system such as muting microphones or routing the signal generator through the loudspeaker system.

Phone Won’t Auto Hang Up

Depending on the revision of the firmware, the SoundStructure device supports auto hang-up from either loop drop detection or call progress detection.

Loop drop detection happens when the central office or the local PBX indicates the remote caller has hang-up by interrupting the loop current or reversing the polarity. Loop drop detection is not always supported by PBX’s.

Call progress detection happens when a busy or fast busy tone is detected as an input signal from the telephone line. The tones are typically generated by the central office or by the local PBX after some period of time after the remote phone participant has hung up.

I Dial But I Don’t Hear The Digits

In SoundStructure, the phone must be taken offhook before the digits will be sent to the telephone interface. In Vortex the phone would go offhook automatically when digits were dialed, but in SoundStructure the phone_connect command must be explicitly sent to take the phone offhook before dialing.

Ethernet

How Do I Determine The IP Address Of My SoundStructure Device?

By default the SoundStructure device has DHCP enabled and will accept an IP address from a DHCP server. A static IP address may also be configured for the SoundStructure device.

It is possible to determine the IP address for the system via several methods:

Connect to the SoundStructure device via RS-232, open SoundStructure Studio and autoscan the device. The IP address will be shown in the Wiring page.

Open DOS shell and ping the network with the broadcast address: xxx.yyy.zzz.255 and then look for the MAC address in the results generated from an ‘arp –a’ command. The MAC address of the SoundStructure device is available from the front of the device inside the front-panel door.

SoundStructure Studio Can’t Find My SoundStructure Device Over Ethernet

Depending on network router configurations, SoundStructure Studio may only be able to find devices that are connected to the same subnet as the local PC that is running SoundStructure Studio. Ensure your PC or control system is on the same subnet as the SoundStructure device.

If on the same subnet and you still can’t find the SoundStructure device with SoundStructure Studio, make sure the SoundStructure device is connected to the ethernet and has either received an IP address from a DHCP server, or has a static IP address that has been set and doesn’t conflict with any other devices on the network. If the DHCP lease has expired or the IP address has changed, it make take a minute or so for the SoundStructure Studio to be able to find the SoundStructure device.

Hardware Troubleshooting

SoundStructure devices have built-in diagnostics that are designed to isolate configuration issues from hardware issues. If the system is not operating according to expectations, the first step is to check the SoundStructure front-panel LED.

The SoundStructure front-panel LED indicates the status of the device as shown in following table. The different states of the SoundStructure front-panel LED are shown in this table.

LED	Color	State	Description
Status	Green	Flashing	The system is starting up.
		Solid	The system is operating normally.
	Yellow	Solid	In an multi-device system, this means that the devices do not have a configuration file that matches the equipment. Upload a valid project using devices that match the actual devices. In other applications, this means the system has logged a warning and the system logs should be reviewed.
	Red	Solid	A system component has failed and requires immediate attention.

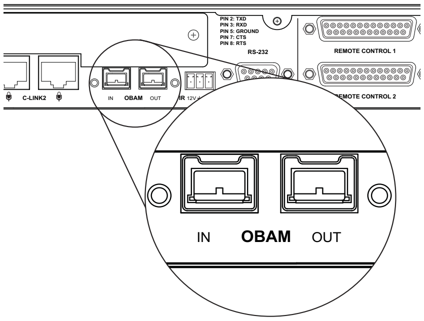
If the front-panel LED is green, then the hardware is operating correctly.

If there is a yellow LED on the front-panel, there is information in a SoundStructure system log that should be reviewed. The LED could be yellow for a variety of reasons including the design file expects a specific device configuration which is not found. An example would be a telephony plug-in card is expected but there isn’t one installed in the device.

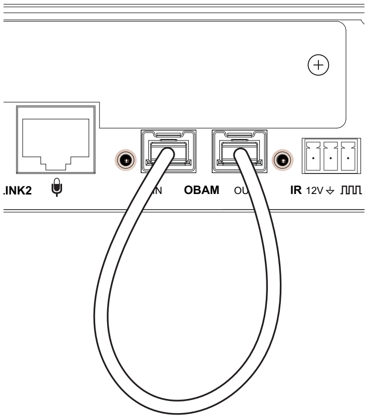
If there is a red LED on the front-panel, it is possible there is a software or hardware issue with the device that may require a firmware update. Check the logs and then contact tech support.

OBAM Troubleshooting

There are status LEDs associated with both the OBAM input and output connections. These LEDs are positioned on either side of the OBAM link connections as shown in the following figure. The OBAM Input LED will illuminate when there is a valid OBAM out connection plugged into the OBAM in connection on this device. The OBAM Output LED will illuminate when the OBAM out connection is plugged into a valid OBAM input port on a different device.



In a multi-SoundStructure device system, if the OBAM LEDs are not illuminated, check that the cables are properly seated into the OBAM in and out connectors. If the cables are properly seated, try looping a known good cable into the OBAM in and out ports as shown in the following figure. If the SoundStructure device’s OBAM interface is working properly the LEDs should illuminate.



Troubleshooting The IR Interface

If you are not receiving command acknowledgments from the IR transmitter, make sure the IR transmitter is sending commands. One easy way to test this is to point the IR transmitter at a video camera and see if the IR transmissions light up on the display screen.

The next step is to make sure the IR receiver is wired properly and terminated to the IR receive port on the SoundStructure rear-panel as shown in the following figure.

By default the SoundStructure device is configured for the Polycom IR remote to have the default device ID of 3 for the SoundStructure to detect the IR key presses.

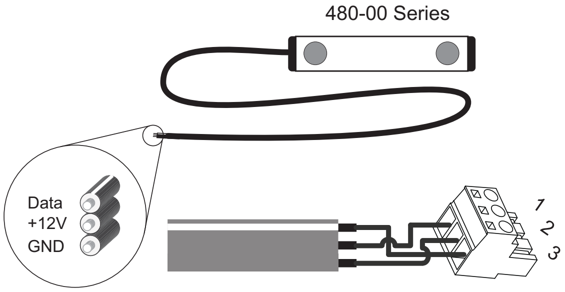
Make sure there is an IR receiver virtual channel defined as follows:

```
vcdef “IR input” control ir_in 1
```

so that when the IR signal is received, there is a command acknowledgment from the IR controller received and reported back as:

```
val ir_key_press “IR Input” 58
```

The key press values returned correspond to the mapping on the Polycom IR remote controller as specified in the Integrator’s Reference Manual for Polycom HDX Systems.



Contacting Technical Support

Before contacting technical support, make sure you have saved the SoundStructure Studio design file and also saved your log file to disk as technical support will want to review these files while helping with the system.

Specifications

Technical Specifications

Dimensions

- 19" (483 mm) W x 13.5" (343 mm) L x 1.75" (45 mm) H (one rack unit)

Weight

- 12 lbs. (5.5 kg) dry, 14 lbs. (6.4 kg) shipping

Connectors

- RS-232: DB9F
- OBAM In/Out: IEEE 1394B
- CLINK2: RJ45
- LAN: RJ45
- Control/Status: DB25F
- Audio: Mini (3.5 mm) quick connect terminal blocks
- IR Receive: Mini (3.5 mm) quick connect terminal block

Power

- Internal power supply
- Input voltage of 90-250 VAC; 50-60 Hz
- Line power requirements (including 0.6 PF): 130 VA (C16), 115 VA (C12), 105 VA (SR12), 95 VA (C8)

Thermal

- Thermal Dissipation (Btu/hr): 266 Btu/hr (C16), 230 Btu/hr (C12), 215 Btu/hr (SR12), 200 Btu/hr (C8)
- Operating temperature 0 - 40° C (104° F)

Operating temperature ranges for the three thermal sensors located on the SoundStructure device are shown in the following table. These sensor values are found on the Wiring page within SoundStructure Studio when connected to a SoundStructure device. Green indicates normal operation up to the temperatures listed in the following table. Yellow indicates an elevated temperature that is acceptable but the ambient temperature and airflow in the system should be checked. Red indicates an over-temperature event that must be corrected for proper operation of the SoundStructure device.

Operating Temperature Ranges for Thermal Sensors on SoundStructure Devices

Sensor	Normal (Green)	Warning (Yellow)	Error (Red)
1	50° C	59° C	60+ ° C
2	69° C	79° C	80+° C
3	53° C	58° C	59+° C

Inputs

- Phantom power: 48 V DC through 6.8 kOhm series resistor per leg, 7.5 mA per channel, software selectable
- Analog input gain: -20 to 64 dB on all inputs in 0.5 dB steps, software adjustable
- Maximum input amplitude: +20.4 dBu, 1% THD + N
- Nominal level: 0 dBu (0.775 Vrms)
- Equivalent input noise: <-122 dBu, 20-20,000 Hz, Rs=150 Ohms (1%)
- Input impedance: 10 kOhms
- Input EMI Filter: Pi filter on all audio inputs

Outputs

- Output gain: -100 to 20 dB in 1 dB steps, software adjustable
- Maximum output amplitude: +23 dBu, 1% THD + N
- Nominal output level: 0 dBu (0.775 Vrms)
- Output impedance: 50 Ohm, each leg to ground, designed to drive loads > 600 Ohms
- Output EMI filter: Pi filter on all audio outputs

System

Note: All Values Valid for All Channels

Unless noted, all values are valid for all channels at 0 dB input gain.

- Frequency response: 20-22,000 Hz, + 0.1 /- 0.3 dB
- Idle channel noise: <-109 dB FS no weighting, 20-20,000 Hz, -60 dB FS, 997 Hz input signal, 0 dB gain
- Dynamic range: >109 dB FS no weighting, 20 - 20,000 Hz, -60 dB FS, 997 Hz input signal, 0 dB gain
- Linearity: 0 dB FS to -122 dB FS +/- 1 dB
- THD+N: < 0.005%, -20 dB FS input signal
- Common mode rejection ratio: <-61 dB, 20-20,000 Hz, no weighting
- Cross talk: <-110 dB, 20-20,000 Hz, 1 kHz, channel-to-channel

- Latency: Mic/Line inputs to outputs: 23 ms, AEC and NC processing enabled
- Acoustic echo cancellation span: 260 ms
- Total cancellation: >65 dB
- Convergence rate: 40 dB/second
- Noise cancellation: 0-20 dB, software selectable
- Control inputs: contact closure
- Status outputs: open collector 60 V and 500 mA maximum total per outputs
- All signal ground pins connected to chassis ground through low impedance planes

Telco

- Input gain: -100 to +20 dB in 1 dB steps, software adjustable
- Nominal transmit level: 0 dBu in SoundStructure device yields -15 to -17 dBm to phone (country code dependent)
- Off hook loop current: 10 mA (minimum) to 120 mA (maximum)
- Output gain: -100 to +20 dB in 1 dB steps, software adjustable
- Frequency response: 250-3300 Hz
- Dynamic range: >70 dB FS, 250-3300 Hz, "A" weighted

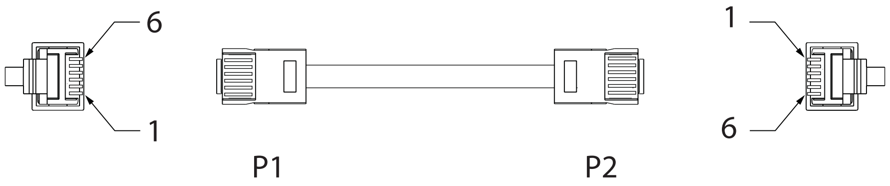
Pin Out Summary

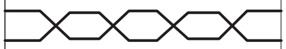
Note: Drawing and Part Numbers For Reference Only

Drawings and part numbers are provided for reference only. Other than cables provided by Polycom, Polycom claims no responsibility or liability for the quality, performance, or reliability of cables based on these reference drawings. Contact a Polycom reseller to order cables that meet the appropriate manufacturing tolerances, quality, and performance parameters for particular applications.

PSTN Cable

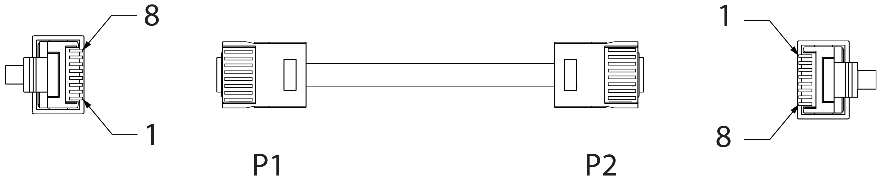
To build a custom telephone cable, use 26AWG twisted-pair cable using the wiring connections shown in the following figure.

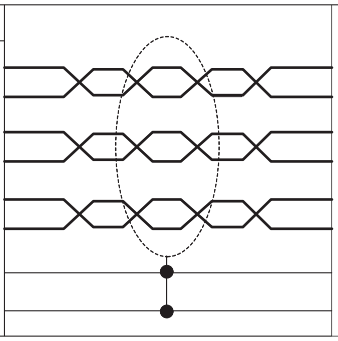


SIGNALS	P1		P2
Unused	1		1
Unused	2		2
RING	3		3
TIP	4		4
Unused	5		5
Unused	6		6

Conference Link2

To build a custom Conference Link2 cable, use shielded CAT5e, or better, and terminate both end connectors, P1 and P2, with standard 8P8C plugs (for example, RJ45) using the wiring connections shown in the following figure. The maximum length for this cable is 100 feet (30 m). Note that this cable provides a cross-over connection between pins 1 and 2 and pins 5 and 6.



COLOR	AWG	P1		P2
WHITE/GREEN	24	1		5
GREEN	24	2		6
WHITE/ORANGE	24	5		1
ORANGE	24	6		2
WHITE/BROWN	24	7		7
BROWN	24	8		8
DRAIN WIRE	24	3		3
SHIELD		SHELL		SHELL

P1 - RJ-45 shielded Keystone jack, L-com RJ110C5-S or equivalent,

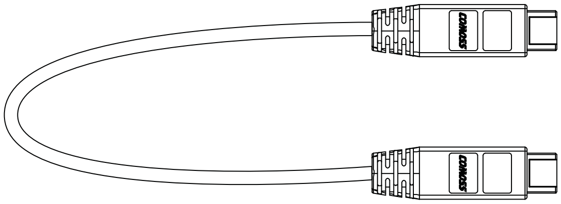
P1 - RJ-45 shielded plug, Tyco 5-569552 or equivalent with shielded RJ-45 panel coupler kit (L-com ECF504-SC5E or equivalent).

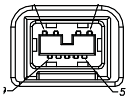
P2- RJ-45 shielded plug, Tyco 5-569552 or equivalent.

OBAM Link

The OBAM cable is a standard 1394b BETA style cable. The current maximum length of this cable is 12 inches.

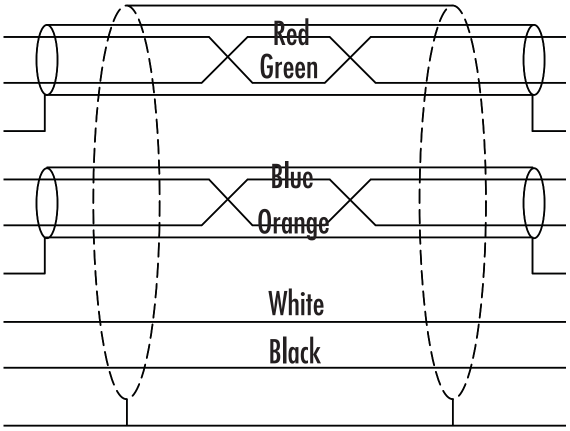
While OBAM Link uses 1394b cables, the underlying bus protocol is not IEEE1394b compliant which means that external IEE1394b devices will not be compatible with OBAM Link. Using IEE1394b hubs or repeaters will not extend the length of OBAM and any non-SoundStructure approved device that is placed on the OBAM Link will prevent OBAM Link from operating properly.





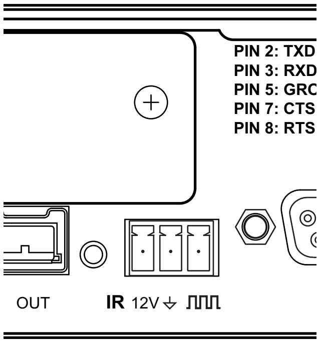
Connector Pinout

Pin 7 is not connected in the below figure.

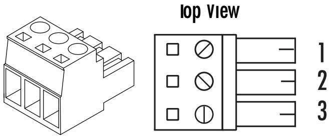
1394b BETA Plug		1394b BETA Plug
1		3
2		4
9		5
3		1
4		2
5		9
6		6
8		8
SHELL		SHELL

IR Receiver

The IR receiver port on the rear-panel of a SoundStructure device is shown in the next figure.


PIN 2: TXD
PIN 3: RXD
PIN 5: GRC
PIN 7: CTS
PIN 8: RTS
OUT
IR 12V ⏚ ⎍⎍⎍

The IR receiver port accepts a standard 3.5 mm terminal block which should be terminated to the IR receiver as shown in the following figures.

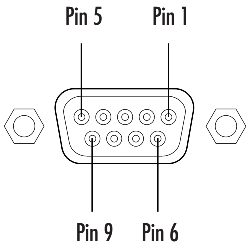

Top View
1
2
3

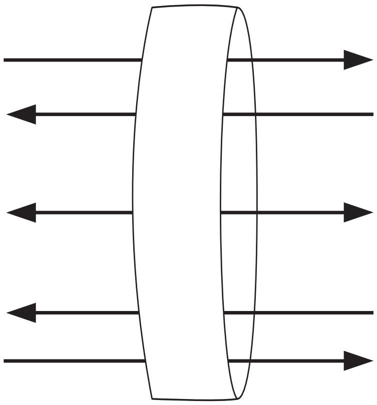
IR Receiver Accepted Terminal Ports

Pin	Signal
1	+12 V
2	Ground
3	IR Signal Data

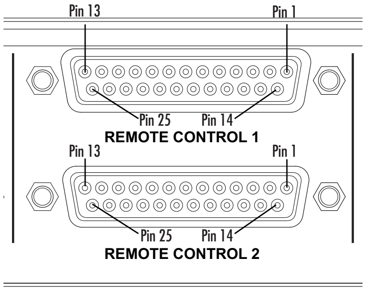
RS-232

The RS-232 interface requires a straight-through cabling to a control system as shown in the following figures.


Pin 5
Pin 1
Pin 9
Pin 6

SoundStructure			Control System	
Pin	Signal		Pin	Signal
1	-		1	-
2	TX		2	RX
3	RX		3	TX
4	-		4	-
5	Ground		5	Ground
6	-		6	-
7	CTS		7	RTS
8	RTS		8	CTS
9	-		9	-

Logic Interface


Pin 13
Pin 1
Pin 25
Pin 14
REMOTE CONTROL 1
Pin 13
Pin 1
Pin 25
Pin 14
REMOTE CONTROL 2

Remote Control Logic Ouput and Input Pin and Signal

Remote Control 1			
Pin	Signal	Pin	Signal
1	+5 V	14	Logic Input 1
2	Logic Output 1	15	Logic Input 2
3	Logic Output 2	16	Logic Input 3
4	Logic Output 3	17	Logic Input 4
5	Logic Output 4	18	Logic Input 5
6	Logic Output 5	19	Logic Input 6

Remote Control Logic Ouput and Input Pin and Signal

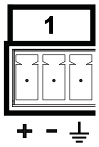
7	Logic Output 6	20	Logic Input 7
8	Logic Output 7	21	Logic Input 8
9	Logic Output 8	22	Logic Input 9
10	Logic Output 9	23	Logic Input 10
11	Logic Output 10	24	Logic Input 11
12	Logic Output 11	25	Ground
13	Analog Gain 1		

Remote Controls Pins and Signals

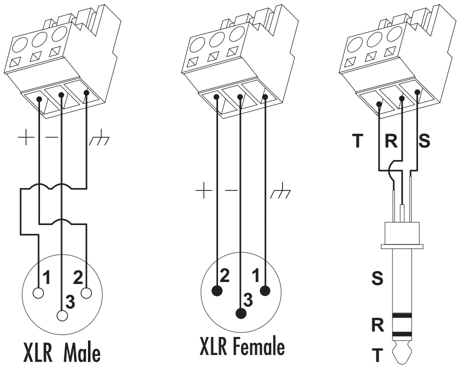
Remote Control 2			
Pin	Signal	Pin	Signal
1	+5 V	14	Logic Input 12
2	Logic Output 12	15	Logic Input 13
3	Logic Output 13	16	Logic Input 14
4	Logic Output 14	17	Logic Input 15
5	Logic Output 15	18	Logic Input 16
6	Logic Output 16	19	Logic Input 17
7	Logic Output 17	20	Logic Input 18
8	Logic Output 18	21	Logic Input 19
9	Logic Output 19	22	Logic Input 20
10	Logic Output 20	23	Logic Input 21
11	Logic Output 21	24	Logic Input 22
12	Logic Output 22	25	Ground
13	Analog Gain 2		

Audio Connections

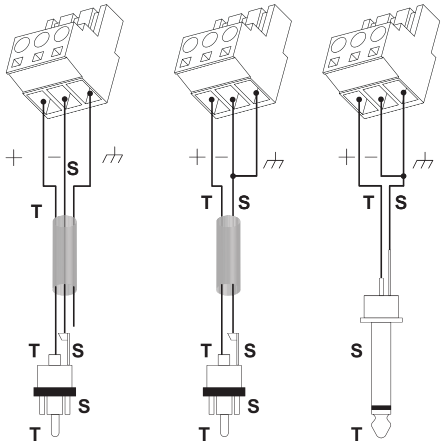
SoundStructure devices provide balanced audio input and output connections that are terminated with 3.5 mm terminal blocks as shown in the following figure.



For each balanced analog input or output on the SoundStructure rear-panel, the first pin should be connected to the positive signal, the second pin is connected to the negative signal, and the third pin is chassis ground as shown in the balanced audio connections in the following figure. To connect the SoundStructure device's audio input and output to other balanced or unbalanced audio equipment, follow the wiring convention in the unbalanced audio connections in the following figure.



Balanced Audio Connections



Unbalanced Audio Connections

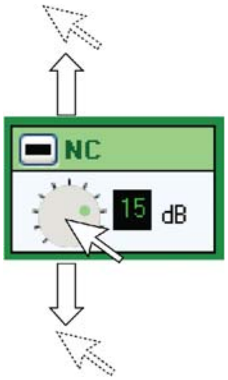
Using SoundStructure Studio Controls

The SoundStructure Studio software environment includes various user interface controls for adjusting the parameters of virtual channels. This section summarizes how to use these controls.

Adjusting Knobs

There are three ways to change the value associated with a knob control:

1. With the mouse: left click (and hold the button) and move the curser up to increase the value and down to decrease the value. Release the mouse when the parameter setting is at the desired value.
2. With the mouse and keyboard: left click on the knob and then use the cursor arrows to change the value by increments of 1 and use the page up and page down commands to move the parameter by 10 dB (or to adjust by octaves) on frequency plots.
3. Keyboard: left click the mouse on the text field and type in a value followed by the Enter key.

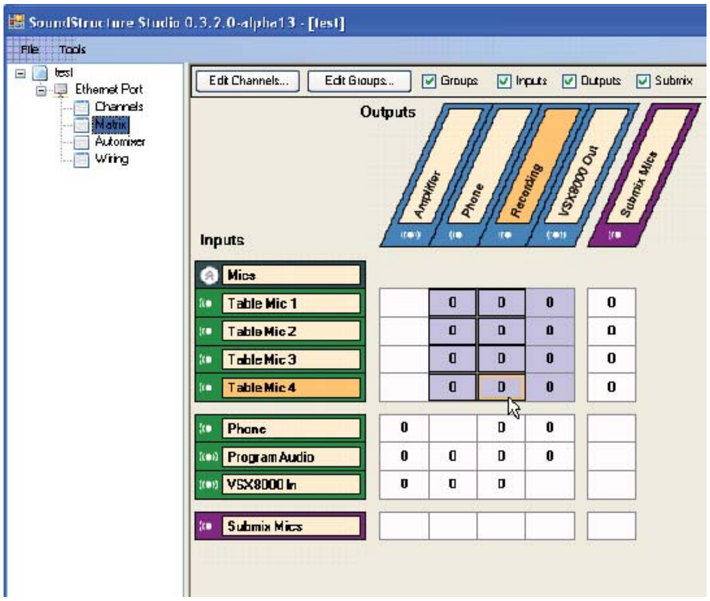


Adjusting Matrix Crosspoints

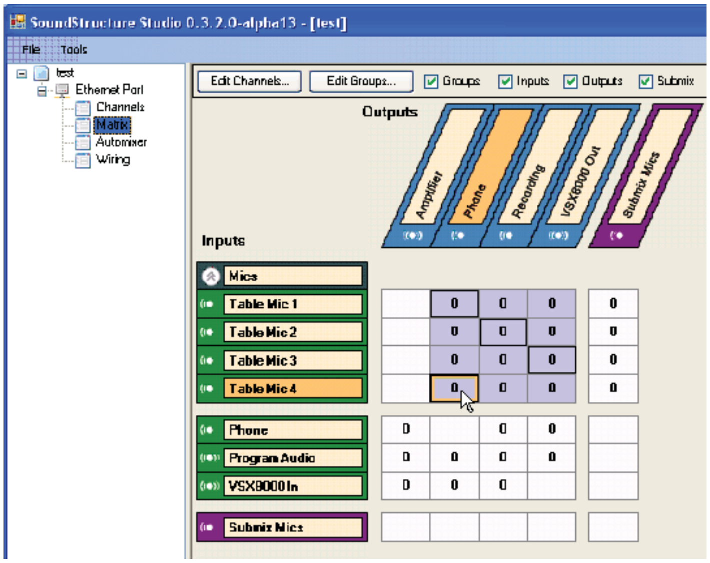
Individual crosspoints can be adjusted by double clicking the crosspoint. This will bring up the matrix control that allows the crosspoint gain, mute status, or which of the three flavors of the input signal to select for this matrix crosspoint.

Multiple crosspoints may be selected in a contiguous area by left clicking on the first cell and dragging across to the bottom cell as shown in the following figure. Once the area is selected, hold down the Control key and double click in any of the cells to bring up the matrix crosspoint control. Any changes made to the control will affect all selected crosspoints.

In addition, an arbitrary collection of crosspoints can be selected by clicking on the first crosspoint and then holding the Control key as other crosspoints are selected. Once the collection of crosspoints has been selected, hold down the Control key and double click any of the cells to bring up the matrix crosspoint control. Any changes made to the matrix control will affect all selected crosspoints.


SoundStructure Studio 0.3.2.0-alpha13 - [test]
File
Tools
test
Ethernet Port
Channels
Matrix
Automixer
Wiring
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Outputs
Inputs
Mics

Inputs	Amplifier	Phone	Recording	VSX8000 Out	Submix Mics
Table Mic 1		0	0	0	0
Table Mic 2		0	0	0	0
Table Mic 3		0	0	0	0
Table Mic 4		0	0	0	0
Phone	0		0	0	
Program Audio	0	0	0	0	
VSX8000 In	0	0	0		
Submix Mics					


SoundStructure Studio 0.3.2.0-alpha13 - [test]
File
Tools
test
Ethernet Port
Channels
Matrix
Automixer
Wiring
Edit Channels...
Edit Groups...
☑ Groups
☑ Inputs
☑ Outputs
☑ Submix
Outputs
Inputs
Mics

Inputs	Amplifier	Phone	Recording	VSX8000 Out	Submix Mics
Table Mic 1		0	0	0	0
Table Mic 2		0	0	0	0
Table Mic 3		0	0	0	0
Table Mic 4		0	0	0	0
Phone	0		0	0	
Program Audio	0	0	0	0	
VSX8000 In	0	0	0		
Submix Mics					

Appendix A: Command Protocol Reference Guide

Using SoundStructure Command Protocols

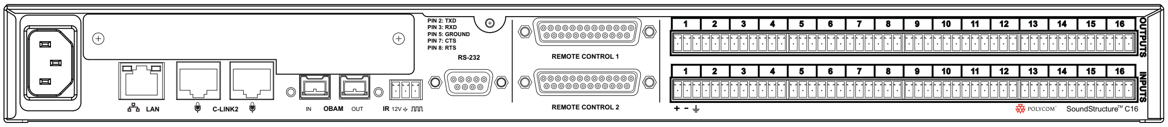
This chapter describes the SoundStructure™ command protocol used to control and configure the SoundStructure products via the RS-232 and Ethernet interfaces. The target audience for this document is the control system programmer and other application developers who need to understand how to control and configure SoundStructure devices.

The purpose of the SoundStructure command and control protocol is to provide an interface for configuring SoundStructure devices and controlling their operating parameters. With SoundStructure devices, a collection of SoundStructure devices linked over OBAM™ will behave as a single device and controlling the collection of devices only requires one connection to a control interface on any of the linked devices.

Understanding SoundStructure Control Interfaces

The SoundStructure control protocol has been designed so that all features are available over all interfaces. Some features will only be practical over the higher bandwidth connections (for example, firmware updates take much less time over the Ethernet interface than the RS-232 interface and signal meters are more responsive over the Ethernet interface). While the SoundStructure Studio Windows software makes full use of the control protocol to configure and control SoundStructure, user applications, such as AMX® and Crestron® control systems will typically only use a subset of the control protocol to adjust settings and monitor system parameters for functions such as muting, volume control, and dialing.

SoundStructure Control Interface

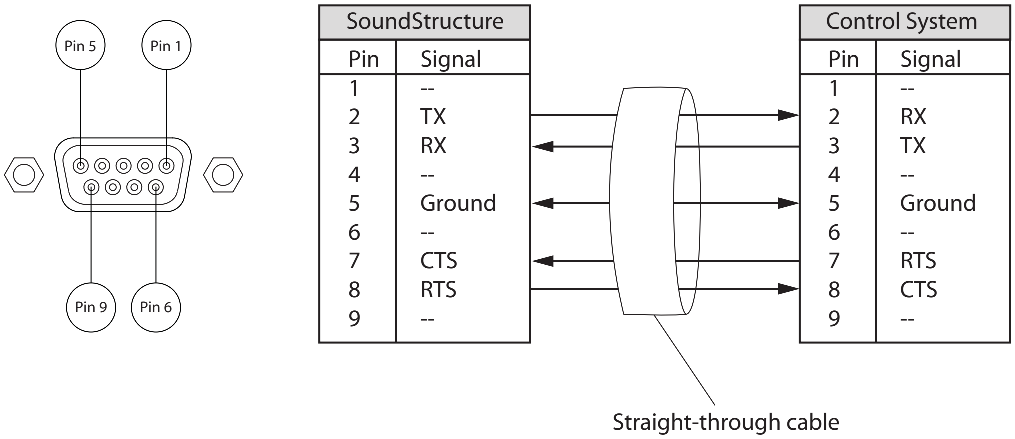


Understanding RS-232

A SoundStructure device's RS-232 port is a female DB9 DCE supporting a fixed data format of eight data bits, no parity, and one stop bit. The supported flow control options are hardware (RTS/CTS) and none. The supported baud rates are 9600, 19200, 38400, 57600, and 115200 with a default baud rate of 9600. This interface is primarily intended for connecting a control system (such as AMX or Crestron) to a SoundStructure device. However, other types of controllers (such as a Windows PC running SoundStructure Studio) may use this interface as well.

The following figure shows the RS-232 pin-out on the rear-panel of the SoundStructure device and requirement for a straight-through cable for connection to an RS-232 port on a control system.

RS-232 Pin Out on SoundStructure System Rear Panel

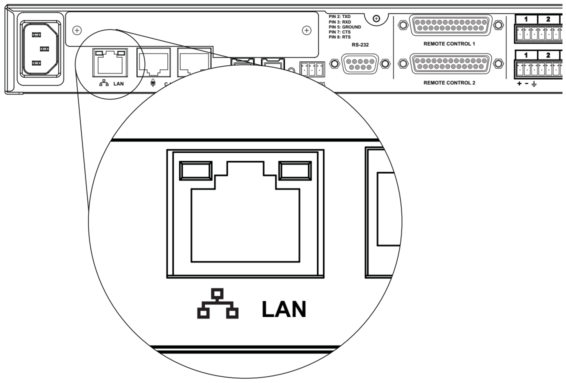

Pin 5
Pin 1
Pin 9
Pin 6
SoundStructure
Pin Signal
1 --
2 TX
3 RX
4 --
5 Ground
6 --
7 CTS
8 RTS
9 --
Control System
Pin Signal
1 --
2 RX
3 TX
4 --
5 Ground
6 --
7 RTS
8 CTS
9 --
Straight-through cable

Connecting with the Ethernet Interface

Each SoundStructure device has a rear-panel Ethernet interface for connecting to the local area network as shown in the following figure. For systems that do not have authentication enabled, connect to the SoundStructure device using port *52774* and telnet communication for systems. There is no administrative login required to interface to SoundStructure devices over port 52774.

For systems that have authentication enabled (see Adding Authentication to SoundStructure Systems), connect to the SoundStructure system using port 52775.

Ethernet Interface on SoundStructure System Rear Panel

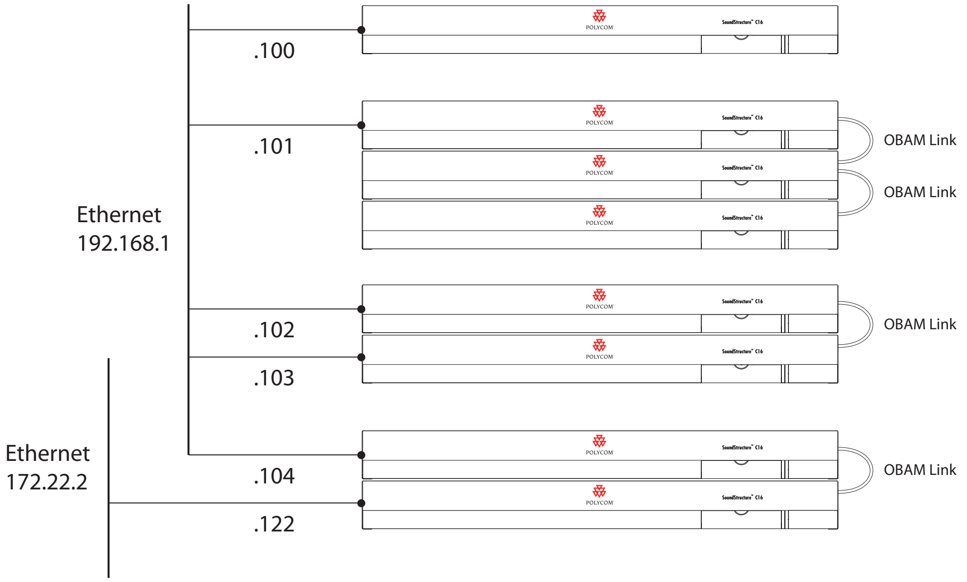

RS-232
REMOTE CONTROL 1
REMOTE CONTROL 2
LAN

Control systems and PCs running SoundStructure Studio may communicate with SoundStructure systems over the Ethernet interface using **port 52774** for open systems and port 52775 for authenticated systems. Each SoundStructure will support multiple simultaneous IP connections from its Ethernet controller.

Each collection of SoundStructure devices that are linked via the OBAM interface only requires a single LAN connection to control all the SoundStructure devices. SoundStructure devices also support having multiple linked devices with each device connected via Ethernet. Connecting to two networks could be used to provide redundancy on the same network or can be used to connect the SoundStructure devices to more than one network.

Multiple network connections can be on the same network or on different subnets as shown in the following figure.

Multiple Network Connections on the Same Network and Different Subnets



The SoundStructure Ethernet interface can be configured to have either a static IP address or can accept a dynamic IP address from a DHCP server. By default the SoundStructure products will accept an IP address from a DHCP server. If there is no DHCP server available, a Link-Local IP address will be created of the form 169.254.abc.def.

Using Virtual Channels

As described in Introducing SoundStructure Design Concepts, a virtual channel is a representation of an individual physical input or output channel. A virtual channel may also be a stereo pair of physical inputs or output channels. The virtual channel name that is created when the virtual channel is defined by the A/V designer is used to refer to that particular input or output instead of using the physical channel number. For example, the designer would define the virtual channel “Podium mic” that is connected, for example, to input physical channel 9 and then refer the virtual channel as “Podium mic”. Once a virtual channel is defined, it is always used to reference that particular signal or signals.

.

Note: Case-Sensitive Virtual Channel Names

The Virtual channel name is case-sensitive: “Podium Mic” and “PODIUM mic” would represent two different virtual channels.

The motivation for using virtual channels is both to allow the control system programming to start before the physical wiring may be known and to make the control system programming re-usable across different installations regardless of how the system is wired. Virtual channels allow third-party control system code to be easily re-used because the controller code controls the SoundStructure devices through the virtual channel names, not the underlying physical input and output that a particular channel is connected to. Virtual channels make the solution more portable and reusable because the control system doesn’t need to know which physical input or output the signal is connected to, it only needs to know the virtual channel name. The use of virtual channels should also improve the quality of the control system code since it is more difficult to confuse “Podium mic” vs. “VCR audio” in the code than it would be to confuse input 7 on device 2 vs. input 9 on device 1. The clarity and transparency of the virtual channel names should reduce the amount of debugging and subsequently reduce the amount of time to provide a fully functional solution.

For instance, if a virtual channel were called “Podium mic” then the control system code would control this channel by sending commands to “Podium mic”. It would not matter to the control system if on one installation “Podium mic” were wired to input 1 and on another installation “Podium mic” was wired to input 7. The same control system code can be used on both installations because the SoundStructure devices would know which underlying physical channel(s) are part of the virtual channel definition. By using the same API commands on different installations that refer to “Podium mic”, the control system code is insulated from the actual physical connections which are likely to change from one installation to the next.

Note: Virtual Channels Controlling and Configuring Physical Channels

Virtual channels are a high-level representation that encompasses information about the physical channel and are used to configure and control the underlying physical channel(s) without having to know which physical input or output the virtual channel is connected to after the virtual channel has been defined.

Within SoundStructure Studio and any third-party controller code, virtual channels are the only way to configure and control the underlying physical channels. The physical input and output channel numbering described in the previous section is used only in the definition of virtual channels so that the virtual channel knows which physical channel(s) it refers to.

A benefit of working with virtual channels is that stereo signals can be more easily used and configured in the system without having to manually configure both the left and right channels independently. Using virtual channels that represent stereo physical signals reduces the chance of improper signal routings and processing selections. The result is that both designs and installations can happen faster and with higher quality.

Understanding Virtual Channel Types

Virtual channels are operated on by the command set which can apply parameter changes to the underlying physical channels. For example, setting the fader parameter of a virtual channel would set the fader parameter for its underlying physical channels.

There are two types of virtual channels in SoundStructure: mono virtual channels and stereo virtual channels.

Understanding Mono Virtual Channels

Mono virtual channels are a representation of a single physical channel. All parameters of the physical channel are controlled through the virtual channel. An example of where a mono virtual channel would be used is a microphone input.

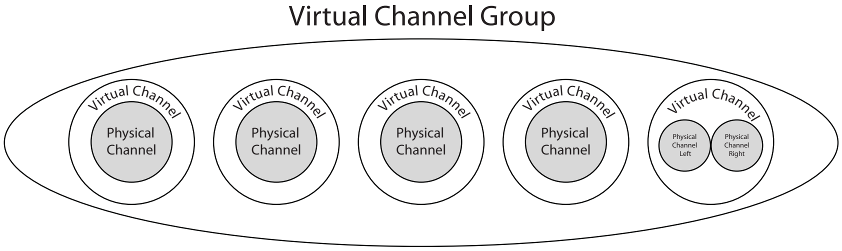
Understanding Stereo Virtual Channels

Stereo virtual channels combine exactly two physical channels to create a stereo pair. All controls and processing take into account the stereo nature of the virtual channel. For example, when mono virtual channels are routed to stereo virtual channels in the matrix, the SoundStructure device will send the mono channel to both stereo channels with the appropriate gain. Additionally, a pan control is available that allows adjustment of the relative signal level in the left and right channels. An example of a stereo virtual channel would be a stereo VCR signal.

Understanding Virtual Channel Groups

It is often convenient to refer to a group of virtual channels and control a group of virtual channels with a single command. Virtual channel groups are used with SoundStructure products to create a single object made up of loosely associated virtual channels. Once a virtual channel group has been created, all commands to a virtual channel group will affect the virtual channels that are defined as part of the virtual channel group and command acknowledgments from all the members of the virtual channel group will be returned. Virtual channel groups may be thought of as a wrapper around a number of virtual channels as shown in the following figure.

Virtual Channel Groups

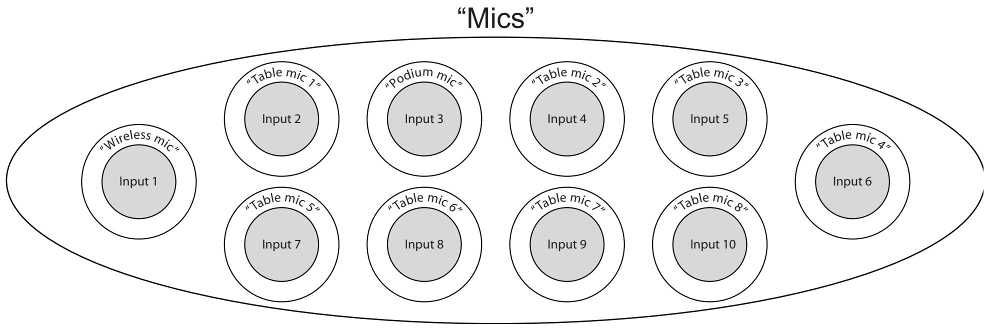


As an example of a virtual channel group, consider in the following figure the creation of the virtual channel group “Mics” made up of the entire collection of individual microphone virtual channels in a room. Once the virtual channel group “Mics” has been created, it is possible to configure and control all the microphones at the same time by operating on the “Mics” virtual channel group.

It is possible to have multiple virtual channel groups that include the same virtual channels. Commands can be sent to the particular virtual channel group will affect the members of the group and those members will respond with the appropriate command acknowledgments.

Note: Virtual Channel Groups including Same Virtual Channels
Multiple virtual channel groups may include the same virtual channels, in other words, a virtual channel can belong to more than one virtual channel group.

Virtual Channel Group



Understanding SoundStructure Command Syntax

The description of the control protocol syntax in this section and elsewhere in this document refers to Unicode characters in four-digit hex format, such as 002A which is the asterisk character in the Basic Latin code page. This is consistent with references such as The Unicode Standard.

The control protocol consists of text-based, human-readable commands and status messages. Binary data transfers are possible (for example, transferring configuration files or sending meter data), but these transfers are initiated with text-based, human readable commands. The commands use the UTF-8 encoding for characters.

Controlling SoundStructure Parameters

The SoundStructure command and control functions allow an external controller to set, query, and monitor parameters of one or more linked SoundStructure devices. There are three types of parameters that can be controlled:

- system parameters,
- virtual channel parameters, and
- matrix parameters.

Understanding System Parameters

System parameters are global and apply to a collection of OBAM-linked SoundStructure devices. A device-specific system parameter affects a parameter on a single SoundStructure device. Examples of device specific system parameters include firmware version and RS-232 baud rate.

Device-specific system parameters are addressed by an integer index that indicates the device ID of the SoundStructure device that is to be controlled. The device ID is created automatically when multiple SoundStructure devices are linked together through the OBAM interface. All stand-alone SoundStructure devices will have a device ID equal to 1. In a multi-device system, the device that has no OBAM in connection is device 1, the device connected to that device is device 2, and so on until the last device. Up to 8 devices may be connected over OBAM.

Understanding Virtual Channel Parameters

Virtual channel parameters are defined for a given virtual channel. Examples of virtual channel parameters include gain, AEC enable, and telephone dialing. These parameters are addressed by the virtual channel name that identifies the virtual channel to be controlled.

Understinading Matrix Parameters

Matrix parameters are defined at crosspoints of the SoundStructure matrix mixer. Since a matrix crosspoint is defined by an input and an output, matrix parameters are addressed by two virtual channel names that identify the input and output virtual channels that define the crosspoint to be controlled.

Parameters can have access modes of *read/write*, *read-only*, or *write-only*. Some parameters can also support user-definable minimum and maximum limits such as volume control commands.

All parameter control commands operate on a specific type of parameter. The parameter types supported by the SoundStructure control protocol are:

void

Void commands take no argument, and must be write-only. For example, the sys_reboot parameter is a write-only void parameter that reboots the SoundStructure device when the command is executed.

boolean

Boolean parameters take one of two values: 0 or 1.

integer

Integer parameters represent an integer value. When incremented or decremented beyond their range, they saturate to their maximum or minimum value, respectively. Integer parameters can support a user-defined minimum and maximum.

float

Float parameters represent a floating-point value. When incremented or decremented beyond their range, they saturate to their maximum or minimum value, respectively. Float parameters can support a user-defined minimum and maximum.

sequence

Sequence parameters represent unsigned integer values. When incremented or decremented beyond their range, they wrap around to their minimum or maximum value, respectively. Sequence parameters do not support a user-defined minimum or maximum.

string

String parameters represent a string value.

list

List parameters represent a sequence of string values. For example, the pstn country parameter is a list parameter that sets the country code for the PSTN telephony interface. Some possible values for the pstn country parameter might be: north america, europe, and china. Even though list parameters are represented as strings, their values are a sequence in a pre-defined order. Thus, they can be incremented and decremented. When incremented or decremented beyond their range, they wrap around to the beginning or end of the list, respectively.

Understanding the Command Format

Referring to the command hierarchy below, each sub-category of command inherits the syntax of its parent and adds further syntax requirements. Starting at the root of the hierarchy, all commands have the following syntax:

<action> <data> <term>

where <action> specifies the system-defined action, <data> is the action-specific arguments or payload data, and <term> is the command terminator.

Actions

The <action> field, also known as the command action, consists only of *lowercase characters*. The full set of actions is provided later in this chapter.

Data

The content and format of the command data is specific to the command action. The SoundStructure control protocol defines three primary groups of actions: channel definition actions, parameter control actions, and data transfer actions. Details on specific command actions are given in later sections.

Command Termination

Commands sent to SoundStructure must be terminated by either a single carriage return (000D) or a carriage return followed by a line feed (000A). The single carriage return is the preferred method of command termination; however both formats will be supported in order to provide a protocol that is robust to differing line end conventions.

Commands (for example, acknowledgments) generated by SoundStructure will always be terminated with a single carriage return (000D).

Note: Lowercase SoundStructure Commands

All commands for SoundStructure must be lowercase and terminated with a single carriage return (000D) or a carriage return (000D) followed by a line feed (000A).

Command Acknowledgments

All commands generate acknowledgments. The format of the acknowledgment and whether it is sent to the originating interface or all interfaces depends on the specific command. In general, the acknowledgment is similar to the command that caused it.

The acknowledgment is sent to all interfaces if a setting changed. The acknowledgment is only sent to the originating interface if no settings changed, for example, a query for a parameter is made.

Command Length

All commands must be less than or equal to 2048 bytes in length, including the terminator.

Understanding the Control Commands

Most of the commands in the SoundStructure control protocol fall under the category of control commands. All control commands have the following syntax:

```
<action> [<arg> [<arg> [<arg> ... ]]]<term>
```

where <action> specifies the system-defined command action and the <term> field is the command terminator.

The <arg> fields comprise the <data> portion of the command. They contain zero or more arguments specific to the given command action. The [and] characters are not present in the actual command, they are used here to indicate that the <arg> parameters are optional, depending on the requirements of the given command action.

General requirements for the syntax of the <arg> fields are given in the following subsections. Specific requirements for the <arg> fields are given in previous sections which describe the syntax of channel definition commands and parameter commands, respectively.

Argument Separation

Control commands sent to SoundStructure must have all <arg> parameters separated by one or more space (0020) or tab (0009) characters. Using a single space is preferred, but SoundStructure supports multiple space and tab characters.

Control commands generated by SoundStructure have all <arg> parameters separated by exactly one space character.

All arguments of control commands will be one of the following types: integer arguments, floating-point arguments, system-defined text arguments, or user-defined text arguments.

Integer Arguments

Integer arguments represent an integer value. They are represented using a string of digits (0030-0039) with an optional leading plus symbol (002B) or minus symbol (002D). Examples of valid integer arguments are 5, -2, and +7. Integer arguments must be less than or equal to 32 bytes in length.

Floating-Point Arguments

Floating-point arguments represent a floating-point value. They are represented using a string of digits (0030-0039), an optional decimal point symbol (002E), an optional E (0045) or e (0065) for indicating an exponent, and optional plus symbols (002B) or minus symbols (002D) for indicating the sign of the mantissa or exponent.

Examples of valid floating-point arguments are 0.618, -4.8, 2, +3.14, 6.022e23, 6.626E-34, and -1.759e11. Floating-point arguments must be less than or equal to 32 bytes in length.

System-Defined Text Arguments

Text arguments that are defined by the command set consists only of digits (0030-0039), lower-case characters (0061-007A), and the underscore character (005F). The underscore character is used when it would make long arguments more readable. Examples of valid system-defined text arguments are cr_mic_in and agc_rate. System-defined text arguments must be less than or equal to 32 bytes in length.

User-Defined Text Arguments

Text arguments and data that are user-defined (for example, virtual channel labels) support all UTF-8 symbols except the control symbols (0000-001F). The full range of UTF-8 symbols is supported to allow user-definable labels in other languages. The control symbols are not supported because they are typically unprintable. In particular, the line feed (000A) and carriage return (000D) symbols are not allowed for two reasons: first, those symbols are used as command terminating characters; and second, the command set does not support the concept of multi-line text arguments.

User-defined text arguments are delimited by a quotation mark symbol (0022) at the start and end of the string. Quotation mark symbols appearing within the text argument can be escaped by a preceding backslash symbol (005C). Literal backslash symbols appearing within the text argument are escaped by a preceding backslash symbol.

Examples of valid user-defined test arguments are “Table Mics”, “Mic 1\\3”, and “\”Program\” Audio”.

User-defined text arguments must be less than or equal to 256 bytes in length. Note that this may be less than 256 symbols, since most of the UTF-8 symbols are multi-byte. The quotation mark delimiters and escape characters are included in the 256 byte limit.

Acknowledgments

Control commands generate acknowledgments that are similar to the command format. The acknowledgments are typically prefixed with the keyword *val* to indicate the value returned from the command.

Understanding Virtual Channel Definition Commands

Virtual channel definition commands are a type of control command that provide methods for defining virtual channels and mapping them to physical channels. The SoundStructure Studio software will create the virtual channel definitions based on the input and output selections the designer has chosen. The syntax described below is what SoundStructure Studio uses to create the channel definitions.

Channel definition commands support the following three actions.

`vcdef`

Define a new virtual channel and its physical channel mapping.

`vcundef`

Delete the definition of a virtual channel.

`vcrename`

Rename a virtual channel.

The syntax for each of these actions is given in the following sections.

vcdef Action

The vcdef action is a virtual channel definition command that defines a new virtual channel and its physical channel mapping. Commands with the vcdef action have the following syntax.

```
vcdef <label>  <vctype>  <pctype>  <num> [<num> ... ]<term>
```

Each of the command arguments is described in the following section.

`<label>`

The `<label>` argument is a user-defined text argument that defines the name for the new virtual channel. If a virtual channel or virtual channel group already exists with the same label, then SoundStructure will respond with an error message.

<vctype>

The `<vctype>` argument is a system-defined text argument that defines the type of virtual channel that will be created. The following values are supported.

Virtual channel type	Description
mono	A mono virtual channel
stereo	A stereo virtual channel
control	A control channel such as logic input or output or IR receiver
control_array	A collection of control pins, in other words a group of logic input or output pins

<pctype>

The `<pctype>` argument defines the physical channel type of the physical channels in the virtual channel. The `<pctype>` argument is a system defined text argument that must be one of the following.

Physical channel type	Description
cr_mic_in	The physical channel is one of the mic/line inputs on a conferencing device (for example, the physical channel supports echo cancellation).
cr_line_out	The physical channel is one of the line outputs on a conferencing device.
sr_mic_in	The physical channel is one of the mic/line inputs on a sound-reinforcement device (for example, the physical channel does not support echo cancellation).
sr_line_out	The physical channel is one of the line outputs on a sound reinforcement device.
pstn_in	The physical channel for the receive signal to the analog telephony interface.
pstn_out	The physical channel for the transmit signal to the analog telephony interface
voip_in	The physical channel for the receive signal to a VoIP telephony interface.
voip_out	The physical channel for the transmit signal from the VoIP telephony interface
sig_gen	The physical channel is the signal generator input.
submix	The physical channel is one of the sub-mix channels.
clink_in	The physical channel is one of the ConferenceLink inputs.

Physical channel type	Description
clink_out	The physical channel is one of the ConferenceLink outputs.
digital_gpio_in	The physical channel for the digital logic input pins
digital_gpio_out	The physical channel for the digital logic output pins
analog_gpio_in	The physical channel for the analog logic input pins
ir_in	The physical channel for the infrared remote control port

<num>

One or more `<num>` arguments are required to define the global channel index (indices) of the physical channel(s) in the virtual channel. The `<num>` argument is an integer argument.

As an example, consider two SoundStructure C16 devices linked via OBAM link. The following command defines a stereo virtual channel consisting of the last microphone on the first device and the first microphone of the second device.

```
vcdef “Stereo Mics” stereo mic_in 16 17
```

Since this virtual channel type is stereo, an even number of <num> arguments must be specified, otherwise an error message will be generated.

The following command creates a logic input pin called “logic input” that is on logic pin 1

```
vcdef “logic input” control digital_gpio_in 1
```

For mono virtual channels, a single `<num>` argument must be specified; otherwise an error message will be generated. For stereo virtual channels, two `<num>` arguments must be specified; otherwise an error message will be generated. The first `<num>` argument corresponds to the left channel, and the second corresponds to the right channel. For `control_array` virtual channels more than two `<num>` arguments may be specified as in the following example.

```
vcdef "logic array" control_array digital_gpio_in 2 3 4
```

which creates a logical group using logic inputs 2, 3, and 4.

To create a channel that can report IR commands:

```
vcdef “ir receiver” control ir_in 1
```

This creates the virtual channel name “ir receiver” that will report back any IR key presses that are received using the standard Polycom IR receiver that has been set to a device ID of 3.

vcdef Acknowledgments

When a virtual channel definition command with the vcdef action is successfully executed, SoundStructure will send an acknowledgment in the same format as the command. The acknowledgment will be sent to all interfaces.

As an example, consider two C16 linked via OBAM link, and assume that no virtual channels are defined. If a control system connected to any of the control interfaces of a SoundStructure device sends the following command:

```
vcdef "Stereo Mics" stereo mic_in 16 17
```

then the following acknowledgment will be generated and sent to all control interfaces.

```
vcdef "Stereo Mics" stereo mic_in 16 17
```

As an example of creating a monaural microphone connected to input 8:

```
vcdef “Podium mic” mono cr_mic_in 8
```

And the system will respond with

```
vcdef “Podium mic” mono cr_mic_in 8
```

vclist Action

The vclist action returns the complete list of virtual channels that have been defined with the vcdef action. The vclist action accepts no arguments and has the following syntax:

```
vclist
```

vclist Acknowledgments

When the vclist command is executed, SoundStructure will send the acknowledgment prefaced with vcitem in the following syntax:

```
vcitem <label> <vctype> <pctype> <num> [<num> ...] <term>
```

A vcitem acknowledgment will be received for each virtual channel that has been defined. The acknowledgment will be sent to the interface that initiated the request.

Each of the acknowledgment arguments is defined below.

<label>

The <label> argument is a user-defined text argument that specifies the name of the virtual channel that was defined.

<vctype>

The <vctype> argument is a system-defined text argument that defines the type of virtual channel that are created. The list of vctypes is included in the vcdef Action section.

<pctype>

The <pctype> argument defines the physical channel type of the physical channels in the virtual channel. The <pctype> argument is a system defined text argument that must be one of the pctypes listed in the vcdef Action section.

<num>

One or more <num> arguments are returned with the indices of the physical channel(s) defined as part of the virtual channel.

vcundef Action

The vcundef action is a virtual channel definition command that undefines a virtual channel that was previously defined with the vcdef action. Commands with the vcundef action have the following syntax.

```
vcundef <label> <term>
```

Each of the command arguments is defined below.

<label>

The <label> argument is a user-defined text argument that defines the name of the virtual channel to be undefined. If no virtual channel exists with the given label, then SoundStructure will respond with an error message.

vcundef Acknowledgments

When a virtual channel definition command with the vcundef action is successfully executed, SoundStructure will send an acknowledgment in the same format as the command. The acknowledgment will be sent to all interfaces.

As an example, consider a SoundStructure system that has a virtual channel defined with “Stereo Mics” as its label. If the following command is sent to the SoundStructure system,

```
vcundef "Stereo Mics"
```

then the following acknowledgment will be generated and sent to all interfaces.

```
vcundef "Stereo Mics"
```

vcrename Action

The vcrename action is a virtual channel definition command that changes then name of a virtual channel. Commands with the vcrename action have the following syntax.

```
vcrename <label>  <new-label>
```

Each of the command arguments is defined below.

<label>

The <label> argument is a user-defined text argument that specifies the name of the virtual channel to be renamed. If no virtual channel exists with the given label, then the SoundStructure device will respond with an error message.

<new-label>

The <new-label> argument is a user-defined text argument that specifies the new name to assign to the virtual channel. If a virtual channel or virtual channel group already exists with the same label, then the SoundStructure device will respond with an error message.

vcrename Acknowledgements

When a virtual channel definition command with the vcrename action is successfully executed, SoundStructure will send an acknowledgement in the same format as the command. The acknowledgement will be sent to all interfaces.

Virtual Channel Group Definition Commands

Virtual channel group definition commands are a type of control command that provide methods for defining virtual channel groups. Virtual channel group definition commands support the following six actions.

vcgdef

Define a new virtual channel group.

vcgundef

Delete a virtual channel group definition.

vcgrename

Rename a virtual channel group.

vcgadd

Add a virtual channel member to a virtual channel group.

vcgremove

Remove a virtual channel member from a virtual channel group.

vcglist

List the members of a virtual channel group.

vcgdef Action

The vcgdef Action is a virtual channel group definition command that defines a new virtual channel group. The action may define an empty virtual channel group or it may specify one or more virtual channel labels as members of the virtual channel group. Commands with the vcgdef action have the following syntax.

```
vcgdef <label> [<vcmember> [<vcmember> ... ]]<term>
```

Each of the command arguments is described below.

<label>

The <label> argument is a user-defined text argument that defines the name for the new virtual channel group. If a virtual channel group or virtual channel already exists with the same label, the SoundStructure device will respond with an error message.

<vcmember>

Zero or more <vcmember> arguments may be specified to initialize the virtual channel group with virtual channel members. The <vcmember> argument is a user-defined text argument that defines the name of a virtual channel. If no virtual channel with the specified name exists, the SoundStructure device will respond with an error message.

vcgdef Acknowledgement

When a virtual channel group definition command with the vcgdef action is successfully executed, the SoundStructure device will send an acknowledgement in the same format as the command. The acknowledgement will be sent to all control interfaces.

vcgundef Action

The vcgundef action is a virtual channel group definition command that undefines a virtual channel group that was previously defined with the vcgdef action. Commands with the vcgundef action have the following syntax.

```
vcgundef <label><term>
```

Each of the command arguments is defined below.

<label>

The <label> argument is a user-defined text argument that defines the name of the virtual channel group to be undefined. If no virtual channel group exists with the given label, then the SoundStructure device will respond with an error message.

vcgundef Acknowledgement

When a virtual channel group definition command with the vcgundef action is successfully executed, the SoundStructure device will send an acknowledgement in the same format as the command. The acknowledgement will be sent to all control interfaces.

vcgrename Action

The vcgrename action is a virtual channel group definition command that changes the name of a virtual channel group. Commands with the vcgrename action have the following syntax:

```
vcgrename <label> <new-label><term>
```

Each of the command arguments is described below.

<label>

The <label> argument is a user-defined text argument that specifies the name of the virtual channel group to be renamed. If no virtual channel group exists with the given label, then the SoundStructure device will respond with an error message.

<new-label>

The <new-label> argument is a user-defined text argument that specifies the new name to assign to the virtual channel group. If a virtual channel group or virtual channel already exists with the same label, then the SoundStructure device will respond with an error message.

vcgrename Acknowledgements

When a virtual channel group definition command with the vcgrename action is successfully executed, the SoundStructure device will send an acknowledgement in the same format as the command. The acknowledgement will be sent to all interfaces.

vcgadd Action

The vcgadd action is a virtual channel group definition command that adds a virtual channel member to a virtual channel group. Commands with the vcgadd action have the following syntax.

```
vcgadd <label> <vcmember><term>
```

Each of the command arguments is described below.

<label>

The <label> argument is a user-defined text argument that specifies the name of the virtual channel group to which the new member will be added. If no virtual channel group exists with the given label, then the SoundStructure device will respond with an error message.

<vcmember>

The <vcmember> argument is a user-defined text argument that defines the name of the virtual channel to be added to the virtual channel group. If no virtual channel with the specified name exists, the SoundStructure device will respond with an error message. If the virtual channel is already a member of the virtual channel group, the SoundStructure device will respond with an error message.

vcgadd Acknowledgements

When a virtual channel group definition command with the vcgadd action is successfully executed, the SoundStructure device will send an acknowledgement in the same format as the command. The acknowledgement will be sent to all control interfaces.

vcgremove Action

The vcgremove action is a virtual channel group definition command that removes a virtual channel member from a virtual channel group. Commands with the vcgremove action have the following syntax.

```
vcgremove <label> <vcmember><term>
```

Each of the command arguments is described below.

<label>

The <label> argument is a user-defined text argument that specifies the name of the virtual channel group from which the member will be removed. If no virtual channel group exists with the given label, then the SoundStructure device will respond with an error message.

<vcmember>

The <vcmember> argument is a user-defined text argument that defines the name of the virtual channel to be removed from the virtual channel group. If no virtual channel with the specified name exists, the SoundStructure device will respond with an error message. If the virtual channel is not a member of the virtual channel group, the SoundStructure device will respond with an error message.

vcgremove Acknowledgements

When a virtual channel group definition command with the vcgremove action is successfully executed, the SoundStructure device will send an acknowledgement in the same format as the command. The acknowledgement will be sent to all interfaces.

vcglist Action

The vcglist action is a virtual channel group definition command that lists the virtual channel members of a virtual channel group. Commands with the vcglist action have the following syntax.

```
vcglist <label><term>
```

Each of the command arguments is described below.

<label>

The <label> argument is a user-defined text argument that specifies the name of the virtual channel group that will have its members listed. If no virtual channel group exists with the given label, then SoundStructure will respond with an error message.

vcglist Acknowledgements

When a virtual channel group definition command with the vcglist action is successfully executed, SoundStructure will send an acknowledgement with the following syntax:

```
vcglist <label> [<vcmember> [<vcmember ... ]]<term>
```

This acknowledgement uses the same syntax as the vcgdef command, but with the vcglist action. The <label> argument is a user-defined text argument indicating the name of the virtual channel group, and zero or more <vcmember> arguments will indicate the virtual channel members of the virtual channel group.

The acknowledgement will only be sent to the control interface on which the command was received.

As an example, consider a system where we have defined a virtual channel group as follows:

```
vcgdef "all zones" "zone 1" "program audio" "zone 2"
vcgremove "all zones" "program audio"
vcgadd "all zones" "zone 3"
```

If we now send the following vcglist command,

```
vcglist "all zones"
```

Then SoundStructure will send the following acknowledgement to the interface on which the vcglist command was received.

```
vcglist "all zones" "zone 1" "zone 2" "zone 3"
```

Adjusting Parameters

Parameters are adjusted by executing commands on a SoundStructure device. There are three types of commands: system parameter commands, virtual channel commands, and matrix commands. These commands adjust the corresponding parameter type as described previously. The general syntax for all parameter commands is given in this section.

Parameter Command Syntax

All parameter commands have the following syntax.

```
<action> <param> [<limit>] [<chan> [<chan>]] [<index> [<index> ...]] [<arg>]<term>
```

Some examples of parameter commands are given below.

```
get sys_sw_ver 1
set mic_in_gain "DVD Audio" 10
set fader max "DVD Audio" 10
inc fader "DVD Audio" 2
tog aec_en "Mic 1"
set eq_en "Speaker 1" 1
set peq_gain "Speaker 2" 1 -2.5
set matrix_gain "DVD Audio" "Codec Output" 0
authenticate “admin” “456”
run “Power-On”
```

The fields in the command are described below.

<action>

This is a required field that specifies the action for the command. The requirements for the format of this field are given previously. The action must be one of the following values:

Actions and Values

Action	Description
get	get the current value of the parameter
set	set the current value of the parameter; requires the value as an argument
inc	increment the current value; requires the value to increment by as an argument
dec	decrement the current value; requires the value to decrement by as an argument tog toggles the state of the current value; only applicable to Boolean commands
tog	toggles the current value of the boolean parameter
ping	does not affect any parameter, just checks that the system is accessible
run	runs a preset or partial preset
authenticate	logs in to a system that requires authentication. The first parameter to the authenticate action must be the name "admin" and the second parameter is the password.

<param>

This is a system-defined text argument that specifies the name of the parameter on which to operate.

<limit>

Some parameters support user-definable minimum and maximum values. For these commands, the <limit> argument can be specified. The <limit> argument is a system-defined text argument and can be one of the following values:

Limits and Values

Limit	Description
min	operate on the minimum limit for the parameter
max	operate on the maximum limit for the parameter

The behavior of a command when it reaches its minimum or maximum is determined by the parameter type as described previously. This is typically used with the fader parameter.

<chan>

Commands that operate on virtual channels may require one or more channel arguments to define the channel on which to operate. Specifically, system commands require zero or one channel arguments, virtual channel commands require one channel argument, and matrix commands require two channel arguments. For example, the fader virtual channel command requires that a virtual channel or virtual channel group be specified.

<index>

Parameters may be multi-dimensional while most parameters are scalar. For example, the gain parameter is scalar, meaning that there is one value for each physical or virtual channel. An example of a one-dimensional parameter is the parametric EQ gain parameter. There are multiple bands of parametric

EQ for each physical or virtual channel. The <index> arguments are integer arguments used to address parameters with a dimensionality of one or higher. The dimensionality of a command is given in the specific requirements for that command.

<arg>

The meaning of the argument is specific to each parameter. The syntax of an argument is determined by its type. Some uses of commands do not require an argument (for example, get, to get the value of a parameter does not need an argument).

Parameter Modes

Each parameter command enforces one of the following modes for its parameter.

Parameter Modes

Parameter Mode	Description
read-write	the parameter may be both queried and set
read-only	the parameter may be queried, but not set
write-only	the parameter may be set, but not queried

Thus, read-write commands support the get and set actions and support the inc, dec, or tog actions depending on the parameter type. Read-only commands support the get action, but do not support the set, inc, dec, or tog actions. Write-only commands support the set action, but do not support the get, inc, dec, or tog actions.

Parameter Types

All commands fall into one of the following types.

Void

Commands to adjust void parameters that take no arguments. Void parameter commands support the set action. Void parameter commands do not support the get, inc, dec, or tog actions. Void parameter commands must always be write-only. An example of a void command is the sys_reboot command, which performs a software reset when set.

Boolean

A Boolean parameter command’s argument is an integer argument that must be either 0 or 1. Boolean parameter commands support the get, set, and tog actions according to the command’s read-write mode. The tog action causes the parameter to change state (for example, 0 changes to 1, and 1 changes to 0). Boolean parameter commands do not support the inc and dec actions.

Integer

Integer parameter commands control integer-valued parameters with values in the range of -2,147,483,648 (-2^31) to 2,147,483,647 (2^31 - 1), inclusive. The specific command will most likely impose minimum and maximum limits more restrictive than this range. The argument to an integer parameter command is an integer argument.

Integer parameter commands support the get, set, inc, and dec actions according to the command’s read-write mode. Integer parameter commands do not support the tog action. Integer parameter commands may also support user-definable minimum and maximum limits in addition to the system minimum and maximum limits. When performing increment and decrement actions on integer parameter commands, the parameter saturates at the minimum or maximum value (as opposed to wrapping).

Float

Float parameter commands control floating point valued parameters with minimum and maximum limits specific to each command. The argument to a float parameter command is a floating-point argument. Float parameter commands may also support user-definable minimum and maximum limits in addition to the system minimum and maximum limits.

Float parameter commands support the get, set, inc and dec actions according to the command’s read-write mode. When performing increment and decrement actions on float parameters, the parameter saturates at the minimum or maximum value rather than wrapping.

Sequence

Sequence parameter commands control integer-valued parameters with values in the range of 0 to 4,294,967,265 (2^32-1), inclusive. The specific command will most likely impose minimum and maximum limits more restrictive than this range. The argument to a sequence parameter command is an integer argument. Sequence commands do not support user-definable minimum and maximum values. Sequence parameter commands support the get, set, inc, and dec actions according to the command’s read-write mode. Sequence parameter commands do not support the tog action. When performing increment and decrement actions on sequence parameter commands, the parameter wraps rather than saturating. In other words, incrementing one past the maximum will set the parameter to the minimum, and decrementing one past the minimum will set the parameter to the maximum.

String

String parameter commands control string parameters. String parameters are user-defined text arguments and conform to the requirements as defined previously. String parameter commands support the get and set actions according to the command’s read-write mode. String parameter commands do not support the inc, dec, or tog actions.

List

List parameter commands control parameters that correspond to a list of pre-defined strings. The strings are defined in a pre-determined order by the SoundStructure firmware. The string arguments of list parameter commands are system-defined text arguments and are formatted as described previously. List parameter commands support the get, set, inc, and dec actions according to the command’s read-write mode. The inc and dec actions change the parameter’s value to the next or previous string, respectively.

When incrementing or decrementing beyond the end or beginning of the list, the parameter wraps. List parameter commands do not support the tog action.

Acknowledgements

All parameter commands result in acknowledgements from the SoundStructure device. Acknowledgements are generated with the same syntax as the original command except that they will always indicate the val action (for “value”).

Acknowledgments are generated when either a parameter command is issued or a parameter changes value for some other reason. When a parameter command is executed with the get action, the acknowledgment is only sent to the control interface that the parameter command was received from.

Note: Get Action Acknowledgment

When a parameter command is executed with the **get** action, the acknowledgment is only sent to the control interface that the parameter command was received from.

When a parameter command is executed with the set, inc, dec, or tog actions, then if the action results in the parameter changing value, the acknowledgment is sent to all control interfaces on all devices, otherwise (if the action doesn't change the value of the parameter) the acknowledgment is sent only to the control interface that the parameter command was received on. When a parameter changes state for any reason (for example, command execution, logic pin operations, etc.) an acknowledgment is sent to all interfaces on all devices.

As an example, consider the fader command, and assume a Mic Input mono virtual channel has been defined with the label "Microphone 1". Also assume the current value of the fader parameter for that channel is 3. If a control program connected to any control interface of the SoundStructure device sends the following command:

```
set fader "Microphone 1" 6.0
```

then the following acknowledgment will be generated,

```
val fader "Microphone 1" 6.0
```

This acknowledgment will be sent to all control interfaces on all SoundStructure devices. Now, if the control program sends this command:

```
get fader "Microphone 1"
```

then the following acknowledgment will be generated,

```
val fader "Microphone 1" 6.0
```

but this acknowledgment will only be sent to the specific device and control interface that the control program is communicating through.

This implementation of the command protocol has been designed with these frugal acknowledgments because some control systems have limited buffer sizes that are susceptible to buffer overflows when large amounts of data traffic are generated that the particular control port didn't request.

Command List

The complete system parameter command reference is found in the file SoundStructure-parameters.html on the CDROM and may also be found by browsing in the SoundStructure device's web interface by pointing a browser at the IP address of the SoundStructure device.

The commands in this file are organized by the type of command including:

- Gain and Mute
- Matrix
- Telephony
- Equalizer
- Dynamics Processing
- Algorithm
- Input path selection
- Automix
- GPIP Control parameters
- Control Port Parameters
- System Parameters

Command Example

As an example of how to interpret the command information, consider the fader command description below.

Digital Fader

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: -100.0, Maximum: 20.0, Resolution: 0.1
Default	0.0
User Limits Supported	Yes
Event Source	Yes

Description

This parameter sets the fader level (in dB) in the digital domain.

Interpretation of the Arguments

The Channel Type entry indicates that this command accepts a virtual channel name as an argument.

The Value Type entry indicates that this command accepts a floating point number to set the fader level to.

The Read/Write Mode entry indicates that the fader value can be set with the action *set* and the value can be retrieved with the action *get.*

The Phys Chans entry indicates the types of physical channel this command can operate on. The fader command can operate on most physical input signals include the standard conferencing inputs and outputs (for example, the analog mic or line inputs on the rear-panel of the C-series products), the sound reinforcement inputs and outputs (used with the SR12), telephony input and output channels, submix channels, and the ConferenceLink channels to and from the Polycom Video Codec system.

The Virt Chans entry indicates that the command can operate on both mono and stereo virtual channels.

The System Limits entry indicates the maximum and minimum values for the command. In this example the fader may be set to any value between -100 and +20 in increments of 0.1 dB.

The Default value entry indicates the value this parameter will have if not set. It will default to 0 dB in this example.

The User Limits Supported entry indicates whether it is possible to set maximum and minimum values for the fader to limit the values to only be within those ranges. In this example, the fader command can have a minimum and maximum fader value associated with it.

The Event Source entry indicates whether you can use the parameter as a source in an Event. In this example, the fader command can be used as a source in an Event.

Fader Examples

Examples of using the fader commands and their acknowledgments are shown below:

```
set fader "Amplifier" 1
val fader "Amplifier" 1.0

set fader "Amplifier" 10
val fader "Amplifier" 10.0

set fader max "Amplifier" 10
val fader max "Amplifier" 10.0

set fader min "Amplifier" -20
val fader min "Amplifier" -20.0

get fader "Amplifier"
val fader "Amplifier" 10.0

set fader "Amplifier" -40
val fader "Amplifier" -20.0
```

In the last example because the fader min was set to -20, trying to set the fader to -40 limited the value to -20 automatically.

SoundStructure Parameters

Gain and Mute Parameters

Description

The fader, gain, and mute parameters are described here. The telephony gains, faders, and mutes are described in the Telephony Parameters section. The matrix crosspoint gains and mutes are described in the Matrix Parameters section.

Digital Fader

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: -100.0, Maximum: 20.0, Resolution: 0.1
Default	0.0
User Limits Supported	Yes
Event Source	Yes

Description

This parameter sets the fader level (in dB) in the digital domain.

Examples

Command	Response	Description
set fader "Amplifier" 1	val fader "Amplifier" 1	Sets the fader on the "Amplifier" virtual channel to 1.
get fader "Amplifier"	val fader "Amplifier" 1	Returns the current value of the fader on the "Amplifier" virtual channel.
set fader max "Amplifier" 10	val fader max "Amplifier" 10	Sets the maximum fader value to +10 on the "Amplifier" virtual channel. Any commands to set the fader above +10 will have the value set to 10.
set fader min "Amplifier" -15	val fader min "Amplifier" -15	Sets the minimum fader value to -15 on the "Amplifier" virtual channel. Any commands to set the fader below -15 will have the value set to -15.
inc fader "Amplifier" 2	val fader "Amplifier" 3	Increments the current value of the fader by 2dB on the "Amplifier" virtual channel and returns the current value of the fader.
inc fader "Amplifier" 0.5	val fader "Amplifier" 3.5	Increments the current value of the fader by 0.5dB on the "Amplifier" virtual channel and returns the current value of the fader.
dec fader "Amplifier" 2.5	val fader "Amplifier" 1	Decrements the current value of the fader by 2.5dB on the "Amplifier" virtual channel and returns the current value of the fader.

Line Output Gain—`line_out_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output
Virt Chans	Mono, Stereo
System Limits	Minimum: -100.0, Maximum: 20.0, Resolution: 0.5
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the gain (in dB) of the line output.

Examples

Command	Response	Description
set line_out_gain "Amplifier" -10	val line_out_gain "Amplifier" -10	Sets the line_out_gain on the output channel "Amplifier" to -10dB.

Meter Peak Hold Mode—`meter_peak_hold_mode`

Argument	**Argument value**
Channel Type	Global System
Value Type	List
Read/Write Mode	Read/Write
Values	none : Peak Decay hold : One Second Peak Hold (default) reset : Infinite Peak Hold Until Reset
Event Source	No

Description

This parameter defines the peak hold behavior of all of the peak meters in the system. If the infinite peak hold mode is selected, the peaks can be reset using the *meter_peak_reset* parameter.

Meter Peak Reset—`meter_peak_reset`

Argument	**Argument value**
Channel Type	Global System
Value Type	Void
Read/Write Mode	Write-Only
Event Source	No

Description

This parameter resets all of the peak meters in the system, if the peak meters are configured to have the infinite peak hold behavior.

Mic Input Pre-Amp Gain —`mic_in_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
System Limits	Minimum: -20.0, Maximum: 64.0, Resolution: 0.5
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the gain (in dB) of the mic pre-amp. A separate mic/line control is not provided. Instead, a continuous gain range is provided, and the firmware will map this to the appropriate mic/line switch and pre-amp gain settings.

Digital Mute—`mute`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Signal Generator, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	Yes

Description

This parameter sets the mute status of the virtual channel. A value of 0 indicates the virtual channel is unmuted, while a value of 1 indicates it is muted.

Examples

Command	Response	Description
set mute "Amplifier" 1	val mute "Amplifier" 1	Sets the mute on the "Amplifier" virtual channel to true -- the channel is muted.
get mute "Table Mic 1"	val mute "Table Mic 1" 1	Queries the mute status on the "Table Mic 1" virtual channel to see if the channel is muted. The value 1 means it is muted, 0 means it is not muted
set mute "Mics" 1	val mute "Table Mic 1" 1 val mute "Table Mic 2" 1 val mute "Table Mic 3" 1 val mute "Table Mic 4" 1 val mute "Mics" 1	Query the current value of the mute of the virtual channel group "Mics" which has four microphones as group members, "Table Mic 1", "Table Mic 2", "Table Mic 3", and "Table Mic 4".

48 V Phantom Power—`phantom`

Argument	**Argument value**
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Default	0
Event Source	Yes

Description

Enables or disable phantom power on mic inputs. Setting *phantom* to 1 enables phantom power, while setting it to 0 disables phantom power.

Examples

Command	Response	Description
set phantom "Table Mic 1" 1	val phantom "Table Mic 1" 1	Enables the 48V phantom power supply for the input channel "Table Mic 1".

Safety Mute—`safety_mute`

Argument	**Argument value**
Channel Type	Global System
Value Type	Boolean
Read/Write Mode	Read/Write
Default	0
Event Source	Yes

Description

This parameter sets the status of the safety mute. If safety mute is enabled (1), all line outputs of all the devices are muted.

Examples

Command	Response	Description
set safety_mute 1	val safety_mute 1	Enables the safety_mute for a SoundStructure system.
set safety_mute 0	val safety_mute 0	Turns off the safety_mute for a SoundStructure system.

Signal Activity Threshold—`signal_activity_thresh`

Argument	Argument value
Channel Type	Global System
Value Type	Floating-Point
Read/Write Mode	Read/Write
System Limits	Minimum: -100.0, Maximum: 20.0, Resolution: 0.1
Default	-20.0
User Limits supported	No
Event Source	No

Description

This parameter sets the threshold for the signal activity meter.

Gain Trim For Virtual Channels—`trim`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output
Virt Chans	Stereo
Indices	1-32: Physical channel
System Limits	Minimum: -20.0, Maximum: 20.0, Resolution: 0.5
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter applies gain (in the analog domain) to the individual components of a virtual channel. The index indicates to which physical channel of the virtual channel the trim will be applied. For example, indices 1 and 2 correspond to the left and right physical channels of a stereo virtual channel.

Examples

Command	Response	Description
set trim "Program Audio" 1 2	val trim "Program Audio" 1 2	Sets the trim value of the left channel (channel 1) of the stereo virtual channel "Program Audio" to 2dB.
set trim "Program Audio" 2 -3	val trim "Program Audio" 2 -3	Sets the trim value of the right channel (channel 2) of the stereo virtual channel "Program Audio" to -3dB.

Matrix Parameters

Matrix Crosspoint Balance—`matrix_balance`

Argument	Argument value
Channel Type	Matrix
Value Type	Floating-Point
Read/Write Mode	Read/Write
Row Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input, Submix, ConferenceLink Aux Input, ConferenceLink Raw Input
Row Virt Chans	Stereo
Col Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output, Submix, ConferenceLink Aux Output
Col Virt Chans	Mono, Stereo
System Limits	Minimum: -1.0, Maximum: 1.0, Resolution: 0.01
Default	0.0
User Limits Supported	No
Event Source	No

Description

The *matrix_balance* parameter is available at crosspoints where stereo virtual channels are mixed to mono or stereo virtual channels. The *matrix_balance* parameter provides a way to control the amount of gain going to the left and right channels.

Examples

Command	Response	Description
set matrix_balance "Program Audio" "Codec Line Mix Out" 1	val matrix_balance "Program Audio" "Codec Line Mix Out" 1.000	Sends only the right channel of the stereo program audio "Program Audio" source to both stereo audio outputs "Codec Line Mix Out"
set matrix_balance "Program Audio" "Amplifier" 0	val matrix_balance "Program Audio" "Amplifier" 0.000	Sets the balance so that left is sent to left and right is sent to right on the stereo virtual channel input and output. This ensures the stereo program audio "Program Audio" left and right channels are sent to the stereo audio output "Amplifier" left and right channels, respectively.

Matrix Crosspoint Gain—`matrix_gain`

Argument	Argument value
Channel Type	Matrix
Value Type	Floating-Point
Read/Write Mode	Read/Write
Row Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input, Analog Telephony Input Signal Generator, Submix, ConferenceLink Aux Input, ConferenceLink Raw Input
Row Virt Chans	Mono, Stereo
Col Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output, Analog Telephony Output, Submix, ConferenceLink Aux Output
Col Virt Chans	Mono, Stereo
System Limits	Minimum: -100.0, Maximum: 20.0, Resolution: 0.1
Default	0.0
User Limits Supported	Yes
Event Source	No

Description

This parameter sets the gain (in dB) for the specified crosspoint in the matrix mixer.

Examples

Command	Response	Description
set matrix_gain "Table Mic 1" "Phone Out" 3	val matrix_gain "Table Mic 1" "Phone Out" 3	Sets the matrix crosspoint gain from input "Table Mic 1" to output "Phone Out" to 3dB.
get matrix_gain "Table Mic 1" "Phone Out"	val matrix_gain "Table Mic 1" "Phone Out" 3	Queries the matrix crosspoint gain from input "Table Mic 1" to output "Phone Out".

Enable Gated Signal At Crosspoint—`matrix_gate`

Argument	**Argument value**
Channel Type	Matrix
Value Type	Boolean
Read/Write Mode	Read/Write
Row Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Row Virt Chans	Mono, Stereo
Col Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output, Analog Telephony Output, Submix, ConferenceLink Aux Output
Col Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter selects whether the gated (1) or ungated (0) version of the input signal is sent to the output.

Examples

Command	Response	Description
set matrix_gate "Table Mic 1" "Phone Out" 1	val matrix_gate "Table Mic 1" "Phone Out" 1	Tells the matrix to use the automixed version of the input signal "Table Mic 1" when creating the output signal "Phone Out".
set matrix_gate "Table Mic 1" "Phone Out" 0	val matrix_gate "Table Mic 1" "Phone Out" 0	Tells the matrix to use the un-automixed version of input "Table Mic 1" when creating the output signal "Phone Out".

Select Gating Type—`matrix_gate_type`

Argument	**Argument value**
Channel Type	Matrix
Value Type	List
Read/Write Mode	Read/Write
Row Phys Chans	Conferencing Mic/Line Input
Row Virt Chans	Mono, Stereo
Col Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output, Analog Telephony Output, Submix, ConferenceLink Aux Output
Col Virt Chans	Mono, Stereo
Values	conf : Conference Style Gating (default) sr : Sound Reinforcement Style Gating
Event Source	No

Description

This parameter selects the gating style for crosspoints with conferencing inputs. Gating is enabled with the *matrix_gate* parameter.

Examples

Command	Response	Description
set matrix_gate_type "Table Mic 1" "Phone Out" conf	val matrix_gate_type "Table Mic 1" "Phone Out" conf	Tells the matrix to use the conferencing version of the input processing of the input signal "Table Mic 1" when creating the output signal "Phone Out".
set matrix_gate_type "Table Mic 1" "Amplifier" sr	val matrix_gate_type "Table Mic 1" "Amplifier" sr	Tells the matrix to use the sound reinforcement version of the input processing of the input signal "Table Mic 1" when creating the output signal "Amplifier".

Matrix Crosspoint Inversion—`matrix_invert`

Argument	**Argument value**
Channel Type	Matrix
Value Type	Boolean
Read/Write Mode	Read/Write
Row Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input, Analog Telephony Input, Signal Generator, Submix, ConferenceLink Aux Input, ConferenceLink Raw Input
Row Virt Chans	Mono, Stereo
Col Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output, Analog Telephony Output, Submix, ConferenceLink Aux Output
Col Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

Inverts the specified crosspoint in the matrix mixer. Setting *matrix_ invert* to 0 sets the crosspoint to normal polarity; setting *matrix_invert* to 1 inverts the crosspoint polarity.

Matrix Crosspoint Mute—`matrix_mute`

Argument	Argument value
Channel Type	Matrix
Value Type	Boolean
Read/Write Mode	Read/Write
Row Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input, Analog Telephony Input, Signal Generator, Submix, ConferenceLink Aux Input, ConferenceLink Raw Input
Row Virt Chans	Mono, Stereo
Col Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output, Analog Telephony Output, Submix, ConferenceLink Aux Output
Col Virt Chans	Mono, Stereo
Default	1
Event Source	No

Description

Mutes or unmutes the specified crosspoint in the matrix mixer. Setting *matrix_mute* to 0 unmutes the crosspoint; setting *matrix_mute* to 1 mutes the crosspoint.

Examples

Command	Response	Description
set matrix_mute "Table Mic 1" "Phone Out" 1	val matrix_mute "Table Mic 1" "Phone Out" 1	Mutes the crosspoint from input "Table Mic 1" to the output "Phone Out" so "Table Mic 1" will not be heard by the remote participants on "Phone Out".
get matrix_mute "Table Mic 1" "Phone Out"	val matrix_mute "Table Mic 1" "Phone Out" 1	Queries the mute status of the crosspoint from "Table Mic 1" to the output channel apos;Phone Out".

Matrix Crosspoint Pan—`matrix_pan`

Argument	Argument value
Channel Type	Matrix
Value Type	Floating-Point
Read/Write Mode	Read/Write
Row Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input, Analog Telephony Input, Signal Generator, Submix, ConferenceLink Aux Input, ConferenceLink Raw Input
Row Virt Chans	Mono
Col Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output, Submix, ConferenceLink Aux Output
Col Virt Chans	Stereo
System Limits	Minimum: -1.0, Maximum: 1.0, Resolution: 0.01
Default	0.0
User Limits Supported	No
Event Source	No

Description

The *matrix_pan* parameter is available at crosspoints where mono virtual channels are mixed to stereo virtual channels. The *matrix_pan* parameter provides a way to control the amount of gain going to the left and right channels.

Examples

Command	Response	Description
set matrix_pan "Phone In" "Amplifier" 1	val matrix_pan "Phone In" "Amplifier" 1.000	Sends the mono virtual channel "Phone In" to only the right channel of the stereo virtual channel "Amplifier".
set matrix_pan "Phone In" "Amplifier" -1	val matrix_pan "Phone In" "Amplifier" -1.000	Sends the mono virtual channel "Phone In" to only the left channel of the stereo virtual channel "Amplifier".
set matrix_pan "Phone In" "Amplifier" 0	val matrix_pan "Phone In" "Amplifier" 0.000	Sends the mono virtual channel "Phone In" to both the left and right channels of the stereo virtual channel "Amplifier".
set matrix_pan "Table Mic 1" "Codec Stereo Mics Out" -1	val matrix_pan "Table Mic 1" "Codec Stereo Mics Out" -1.000	Sends the mono virtual channel "Table Mic 1" to the left channel of the stereo virtual channel "Codec Stereo Mics Out".

Telephony Parameters

Telephony Parameter Summary

Some telephony API commands operate on the input virtual channel and some operate on the output virtual channel. Below is a table of commands and whether they operate on the input or output channel.

Telephony Parameter Commands

Command	Input channel	Output channel
phone_auto_answer_en	?	
phone_connect		?
phone_dial		?
phone_dial_tone_gain	?	
phone_dtmf_gain	?	
phone_entry_tone_en	?	
phone_exit_tone_en	?	
phone_flash		?
phone_flash_delay		?
phone_redial		?
phone_ring	?	
phone_ring_tone_en	?	

Telephony Parameter Commands

Command	Input channel	Output channel
phone_tone_gain	?	
pstn_auto_hangup_loop_en		?
pstn_country		?
pstn_flash_delay_override		?
pstn_in_gain	?	
pstn_line_voltage		?
pstn_loop_current		?
pstn_out_gain		?

Enable Auto-Answer For Telephony Interface—`phone_auto_answer_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Input
Virt Chans	Mono
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the auto-answer feature for the telephony interface.

Examples

Command	Response	Description
set phone_auto_answer_en "Phone In" 1	val phone_auto_answer_en "Phone In" 1	Sets the phone associated with the virtual channel "Phone In" to autoanswer when the phone rings. Note that the phone in virtual channel name must be used, not the phone output virtual channel name.

Connect Or Disconnect Telephony Interface—`phone_connect`

Argument	**Argument value**
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Output, VoIP Telephony Output
Virt Chans	Mono
Default	0
Event Source	Yes

Description

This parameter sets the connection status of the telephony interface. Setting the *phone_connect* status to 1 connects the call, while setting it to 0 disconnects the call.

Examples

Command	Response	Description
set phone_connect "Phone Out" 1	val phone_connect "Phone Out" 1	Takes the phone output channel "Phone Out" offhook. Note that the phone out virtual channel name must be used, not the phone input virtual channel name.
get phone_connect "Phone Out" 0	val phone_connect "Phone Out" 0	Hangs up the phone line associated with the virtual channel "Phone Out". Note that the phone out virtual channel name must be used, not the phone input virtual channel name.

Dial The Telephony Interface—`phone_dial`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	String
Read/Write Mode	Write-Only
Phys Chans	Analog Telephony Output, VoIP Telephony Output
Virt Chans	Mono
System Limits	Max String Length: 32
Event Source	No

Description

This command dials the specified string of digits on the telephony interface. This parameter can be used to dial one digit at a time or many digits all at once. For the PSTN interface, valid digits are '0' through '9', '*', '#', and ',' (delay). For the SoundStructure VoIP Interface, all characters are valid.

If the telephony interface is not connected (see the *phone_connect* parameter) when this parameter is set, the characters will be stored in a dial buffer. When the telephony interface is eventually connected, the characters stored in the dial buffer shall be dialed. If more than 20 seconds pass after the last phone_dial or phone_connect parameters are sent, then the dial buffer is automatically cleared.

Examples

Command	Response	Description
set phone_dial "Phone Out" "9,18009322774"	val phone_dial "Phone Out" "9,18009322774"	Dials the phone line associated with the virtual channel "Phone Out" with the digit string "9,18009322774". The phone line must be offhook for the digits to be dialed - see phone_connect. Note that the phone out virtual channel name must be used, not the phone input virtual channel name.

Delete previously dialed digit—`phone_dial_backspace`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	Analog Telephony Output, VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

Setting this parameter deletes the most recently added character from the phone_dial dial buffer. If there are no characters in the dial buffer, then setting this parameter has no effect. For the PSTN interface, this parameter only affects the dial buffer when the phone interface is on-hook. This is true for the VoIP interface as well, but the parameter also affects the dial buffer when the interface is off-hook before a call is placed.

Dial Tone Gain—`phone_dial_tone_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Input
Virt Chans	Mono
System Limits	Minimum: -100.0, Maximum: 20.0, Resolution: 0.1
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter controls the gain that is applied to the incoming phone signal when dial tone is present.

Examples

Command	Response	Description
set phone_dial_tone_gain "Phone In"-6	val phone_dial_tone_gain "Phone In" -6	Sets the gain of the dial tone heard in the room for the phone associated with the virtual channel "Phone In" to -6dB when the phone is taken offhook. Note that the phone in virtual channel name must be used, not the phone output virtual channel name.

Telephony Input DTMF Gain—`phone_dtmf_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Input
Virt Chans	Mono, Stereo
System Limits	Minimum: -20.0, Maximum: 20.0, Resolution: 0.1
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the gain (in dB) applied to DTMF tones generated to the local room. To adjust the level of ring tones, entry tones, and exit tones played back into the local room, use the *phone_tone_gain* parameter.

Examples

Command	Response	Description
set phone_dtmf_gain "Phone In"-6	val phone_dtmf_gain "Phone In" -6	Sets the level of the dtmf digits that are heard in the local room from the phone interface associated with the virtual channel "Phone In" to -6dB. Note that the phone in virtual channel name must be used, not the phone output virtual channel name.

Enable Entry Tones for Telephony Interface—`phone_entry_tone_en`

Argument	**Argument value**
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Input, VoIP Telephony Input
Virt Chans	Mono
Default	1
Event Source	No

Description

This parameter enables or disables entry tone generation for the telephony interface. If entry tones are enabled (1), then an entry tone is played whenever the auto-answer feature engages and connects the telephony interface. Entry tones and exit tones (see the *phone_exit_tone_en* parameter) are typically enabled to prevent a caller from entering or exiting a conference unannounced.

Enable Exit Tones For Telephony Interface—`phone_exit_tone_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Input, VoIP Telephony Input
Virt Chans	Mono
Default	1
Event Source	No

Description

This parameter enables or disables exit tone generation for the telephony interface. If exit tones are enabled (1), then an exit tone is played whenever the auto-hangup feature engages and disconnects the telephony interface. Entry tones (see the *phone_entry_tone_en* parameter) and exit tones are typically enabled to prevent a caller from entering or exiting a conference unannounced.

Connect Or Disconnect Telephony Interface—`phone_flash`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	Analog Telephony Output
Virt Chans	Mono
Event Source	No

Description

Setting this parameter disconnects the telephony interface then reconnects it after a short delay. The amount of delay can be configured with the *phone_flash_delay* parameter.

Examples

Command	Response	Description
set phone_flash "Phone Out"	val phone_flash "Phone Out"	Flashes the phone interface associated with the virtual channel "Phone Out" to -6dB. Note that the phone out virtual channel name must be used, not the phone in virtual channel name.

Set Flash Delay—`phone_flash_delay`

Argument	**Argument value**
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Output
Virt Chans	Mono
System Limits	Minimum: 100, Maximum: 5000
Default	100
User Limits Supported	No
Event Source	No

Description

This parameter sets the delay (in milliseconds) for the *phone_flash* parameter. Note that by default, PSTN interfaces use the flash delay determined by their *pstn_country* setting. However, they can use the value of this parameter if the *pstn_flash_delay_override* parameter is set to 1.

Ignore an Incoming Call—`phone_ignore`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	Analog Telephony Output, VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

This parameter causes the incoming call to be ignored. The incoming ring tones will be silenced on the local side, but the remote caller will still hear ringing. Taking the phone offhook when there is an incoming ignored call will cause the incoming call to be answered.

Redial The Last Number On The Telephony Interface—`phone_redial`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	Analog Telephony Output, VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

Setting this parameter causes the last number to be redialed on the telephony interface. The last number is defined as all the digits that were dialed since the telephony interface was last connected (see the phone_connect parameter). If the telephony interface is not already connected, setting this parameter will automatically connect it before dialing.

Reject an Incoming Call—`phone_reject`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	Analog Telephony Output, VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

Setting this parameter causes the incoming call to be rejected. For the SoundStructure VoIP Interface, the call is rejected and immediately sent to voice-mail. For the PSTN interface, the call is terminated by automatically answering it, then immediately hanging up. The audio paths remain muted so that the conference is not interrupted and so that no local conference audio is sent to the incoming caller.

Ring Indicator For Telephony Interface—`phone_ring`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read-Only
Phys Chans	Analog Telephony Input, VoIP Telephony Input
Virt Chans	Mono
Event Source	Yes

Description

This parameter indicates the ringing state for the telephony interface. While the telephony interface is ringing, reading this parameter will return 1. When the telephony interface is not ringing, reading this parameter will return 0.

Acknowledgements for this parameter will be automatically sent whenever this parameter changes state due to a hook flash, auto-answer, or auto-hangup.

Examples

Command	Response	Description
	val phone_ring "Phone In" 1	Returns the value 1 when the incoming phone line associated with the virtual channel “Phone In” is ringing. Note that the phone in virtual channel name must be used, not the phone out virtual channel name.

Select Ring Tone For Telephony Interface—`phone_ring_tone`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Sequence
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Input, VoIP Telephony Input
Virt Chans	Mono
System Limits	Minimum: 1, Maximum: 14
Default	1
Event Source	No

Description

This parameter selects the type of ring tone to be generated for the telephony interface. If ring tones are enabled, the selected tone is played whenever there is an incoming ring signal on the telephony interface.

The parameter values correspond to the following tones.

- 1: normal ring
- 2: low trill
- 3: low double trill
- 4: medium trill
- 5: medium double trill
- 6: high trill
- 7: high double trill
- 8: highest trill
- 9: highest double trill
- 10: beeble

- 11: triplet
- 12: low trill precedence
- 13: ring splash
- 14: silent ring

Enable Ring Tones For Telephony Interface—`phone_ring_tone_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Input
Virt Chans	Mono
Default	1
Event Source	No

Description

This parameter enables or disables ring tone generation for the telephony interface. If ring tones are enabled (1), then a ring tone is played whenever there is an incoming ring signal on the telephony interface.

Telephony Input Tone Gain—`phone_tone_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Input, VoIP Telephony Input
Virt Chans	Mono, Stereo
System Limits	Minimum: -20.0, Maximum: 20.0, Resolution: 0.1
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the gain (in dB) applied to tones generated to the local room. In particular, this gain applies to the ring tone, entry tone, and exit tone. To adjust the level of the DTMF digits played back to the local room, use the *phone_dtmf_gain* parameter.

Enable Auto-Hangup On Loop Drop For PSTN Interface —`pstn_auto_hangup_loop_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Output
Virt Chans	Mono
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the auto-hangup on loop drop feature for the PSTN interface.

Enable Auto-Hangup On Call Progress For PSTN Interface —`pstn_auto_hangup_call_prog_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Output
Virt Chans	Mono
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the auto-hangup on call progress tones feature for the PSTN interface.

Country For PSTN Interface—`pstn_country`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Output
Virt Chans	Mono
Values	argentina : Argentina australia : Australia austria : Austria bahrain : Bahrain belgium : Belgium brazil : Brazil bulgaria : Bulgaria canada : Canada chile : Chile china : China colombia : Colombia croatia : Croatia cyprus : Cyprus czech_republic : Czech Republic denmark : Denmark ecuador : Ecuador egypt : Egypt el_salvador : El Salvador finland : Finland france : France germany : Germany greece : Greece guam : Guam hong_kong : Hong Kong hungary : Hungary iceland : Iceland india : India indonesia : Indonesia ireland : Ireland israel : Israel yemen : Yemen

Argument	Argument value
Values	italy : Italy japan : Japan jordan : Jordan kazakhstan : Kazakhstan kuwait : Kuwait latvia : Latvia lebanon : Lebanon luxembourg : Luxembourg macao: Macao malaysia : Malaysia malta : Malta mexico : Mexico morocco : Morocco netherlands : Netherlands new_zealand : New Zealand nigeria : Nigeria norway : Norway oman : Oman pakistan : Pakistan peru : Peru philippines : Philippines poland : Poland portugal : Portugal romania : Romania russia : Russia saudi_arabia : Saudi Arabia singapore : Singapore slovakia : Slovakia slovenia : Slovenia
Values	south_africa : South Africa south_korea : South Korea spain : Spain sweden : Sweden switzerland : Switzerland taiwan : Taiwan tbr21 : TBR21 thailand : Thailand uae : UAE united_kingdom : United Kingdom usa : USA (default)
Event Source	No

Description

This parameter configures the PSTN interface for operation in a specific country.

Tone duration for DTMF Tones—`pstn_dtmf_tone_duration`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Output
Virt Chans	Mono
System Limits	Minimum: 10, Maximum: 600000
Default	100
User Limits Supported	No
Event Source	No

Description

This parameter controls the duration (in milliseconds) of the tone generated for each DTMF digit.

Override Country Code Flash Delay—`pstn_flash_delay_override`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Output
Virt Chans	Mono
Default	0
Event Source	No

Description

This parameter controls whether or not the flash hook delay is determined by the default *pstn_country* settings (0) or by the *phone_flash_delay* setting (1).

PSTN Input Gain—`pstn_in_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Input
Virt Chans	Mono
System Limits	Minimum: -20.0, Maximum: 20.0, Resolution: 0.5
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the gain (in dB) of the signal coming from the PSTN interface.

Examples

Command	Response	Description
set pstn_in_gain "Phone In" 6	val pstn_in_gain "Phone In" 6	Adjusts the input gain on the phone input to 6dB. Note that the phone in virtual channel name must be used, not the phone out virtual channel name.

PSTN Line Voltage—`pstn_line_voltage`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read-Only
Phys Chans	Analog Telephony Output
Virt Chans	Mono
System Limits	Minimum: -128, Maximum: 128
Event Source	No

Description

This parameter indicates the line voltage (in Volts) of the PSTN interface. The value is valid in both on-hook and off-hook modes. The value can be positive or negative, indicating the polarity of the tip/ring voltage. When the value changes sign, it indicates that a polarity reversal has occurred.

Examples

Command	Response	Description
get pstn_line_voltage "Phone Out"	val pstn_line_voltage "Phone Out" 0	Queries the pstn_line_voltage and returns the measured value in Volts on the phone line "Phone Out". Note that the phone out virtual channel name must be used, not the phone in virtual channel name.

PSTN Loop Current—`pstn_loop_current`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read-Only
Phys Chans	Analog Telephony Output
Virt Chans	Mono
System Limits	Minimum: 0.0, Maximum: 281.6, Resolution: 0.1
Event Source	No

Description

This parameter indicates the loop current (in milliamps) of the PSTN interface. The value is only valid when the interface is off-hook.

PSTN Output Gain—`pstn_out_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Analog Telephony Output
Virt Chans	Mono
System Limits	Minimum: -20.0, Maximum: 20.0, Resolution: 0.5
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the gain (in dB) of the signal going to the PSTN interface.

Answer an Incoming Call—`voip_answer`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

This parameter is used to answer an incoming call while the SoundStructure VoIP Interface is currently in a different call.

Specify a Blind Transfer—`voip_blind`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

This parameter is used along with voip_transfer to make a blind transfer.

VoIP Board Info—`voip_board_info`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Read-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
System Limits	Max String Length: 256
Event Source	No

Description

This parameter returns manufacturing and hardware information about the VoIP plug-in card.

Get Bootblock Software Version—`voip_bootblock_sw_ver`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	String
Read/Write Mode	Read-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
System Limits	Max String Length: 256
Event Source	No

Description

This parameter returns manufacturing and hardware information about the VoIP plug-in card.

Get Bootrom Software Version—`voip_bootrom_sw_ver`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	String
Read/Write Mode	Read-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
System Limits	Max String Length: 256
Event Source	No

Description

This parameter returns manufacturing and hardware information about the VoIP plug-in card.

Select the Active Call Appearance—`voip_call_appearance`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Sequence
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
System Limits	Minimum: 1, Maximum: 24
Default	1
Event Source	No

Description

This parameter selects the currently active call appearance. Some parameters (e.g. phone_connect, phone_dial, phone_redial) operate on the currently active call appearance, as specified by this parameter. Setting this parameter is analogous to selecting a call appearance on the UI of a Polycom VoIP phone.

VoIP plug-in card.

Call Appearance Info—`voip_call_appearance_info`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	String
Read/Write Mode	Read-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Indices	1-24: Call appearance index 1-2: Description line
System Limits	Max String Length: 128
Event Source	No

Description

This parameter reports textual information for the specified call appearance. There are two lines of textual information that can be independently queried via the second index to this parameter. Typically, the two lines of information are the local and remote caller ID or number.

Call Appearance Line Number—`voip_call_appearance_line`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Indices	1-24: Call appearance index
System Limits	Minimum: 1, Maximum: 12
Event Source	No

Description

This parameter reports the line number associated with the specified call appearance.

Call Appearance State—`voip_call_appearance_state`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Indices	1-24 : Call appearance index
Values	free : Free dialtone : Dialtone setup : Setup overlap : Overlap proceeding : Proceeding ringback : Ringback connected : Connected disconnected : Disconnected pre_offering : Pre-Offering offering : Offering ncas_call_transfer : Call Transferred ncas_call_conference : Conference Call ncas_call_hold : Call on Hold ncas_call_held : Call Held ncas_call_conference_hold : Conference Call on Hold pvc : PVC preemption_in_progress : Preemption in Progress pre_dialtone : Pre-Dialtone
Event Source	No

Description

This parameter reports the call appearance state for the specified call appearance. Automatic status messages are generated for this parameter when it changes automatically.

Cancel a Transfer or Conference—`voip_cancel`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

This parameter is used to cancel a transfer or conference.

Start a Conference Call—`voip_conference`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

This parameter is used to start a conference call.

Set Boot Server Option—`voip_dhcp_boot_serv`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
Values	option66 : Option 66 custom : Custom static : Static (default) custom_opt66 : Custom + Option 66
Event Source	No

Description

This parameter controls the boot server option for the VoIP interface.

Set Boot Server Option Number—`voip_dhcp_boot_serv_opt`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
System Limits	Minimum: 0, Maximum: 255
Default	150
User Limits Supported	No
Event Source	No

Description

When voip_dhcp_boot_serv is set to custom, this parameter specifies the DHCP option number in which the VoIP card will look for the boot server.

Set Boot Server Option Type—`voip_dhcp_boot_serv_type`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
Values	ip_address : IP Address (default) string : string
Event Source	No

Description

When voip_dhcp_boot_serv is set to custom, this parameter specifies the type of the DHCP option in which the VoIP card will look for the boot server.

Set Boot Server Option 60 Type—`voip_dhcp_option_60_type`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
Values	rfc3925_binary : RFC3925 binary (default) ascii_string : ASCII string
Event Source	No

Description

This parameter specifies the format for the vendor identifying information used with a DHCP server when DHCP option 60 is enabled.

Set Dial Mode —`voip_dial_mode`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Values	number : Number (digit) dialing (default) url : SIP URL dialing
Event Source	No

Description

This parameter is used to select between number dialing and SIP URL dialing. Whenever the phone_connect parameter for a voip_out channel changes from 1 to 0, the voip_dial_mode parameter shall be reset back to number. An automatic status message shall be generated when this happens.

Enable or Disable Do-Not-Disturb Mode—`voip_dnd`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) do-not-disturb mode on the VoIP plug-in card.

VoIP Interface Ethernet Settings–`voip_eth_settings`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	String
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
System Limits	Max String Length: 256
Default	mode=’dhcp’
Event Source	No

Description

This parameter gets or sets the Ethernet settings for the VoIP interface. The system must be rebooted for the Ethernet settings to take effect.

The format of the string is a comma-separated list of name/value pairs with the name and value separated by an equals character, and the value enclosed in single quotes.

The mode attribute is always required. It must be either dhcp or static.

The addr, dns, gw, and nm attributes are required when mode is static and ignored (not required) when mode is dhcp. They are always returned in the acknowledgement. For the dhcp case, they specify the values obtained from the DHCP server.

The addr attribute specifies the IP address of the interface. The dns attribute specifies the domain name server(s). A single server or multiple servers (separated by spaces) may be specified. The gw attribute specifies the gateway. The nm parameter specifies the netmask.

DHCP Example

```
set voip_eth_settings "VoIP In" "mode='dhcp'"
```

```
val voip_eth_settings "VoIP In" "mode='dhcp',addr='172.22.2.129',dns='172.22.1.1 172.22.1.2',gw='172.22.2.254',nm='255.255.255.0'"
```

Static IP Example

```
set voip_eth_settings "VoIP In" "mode='static',addr='172.22.2.200',dns='172.22.1.1',gw='172.22.2.254',nm='255.255.255.0'"
```

```
val voip_eth_settings "VoIP In" "mode='static',addr='172.22.2.200',dns='172.22.1.1',gw='172.22.2.254',nm='255.255.255.0'".
```

Set VLAN ID for the VoIP Interface—`voip_eth_vlan_id`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
System Limits	Minimum: -1, Maximum : 4096
Default	-1
User Limits Supported	No
Event Source	No

Description

This parameter sets the VLAN ID for the VoIP card. A value of -1 corresponds to "disabled."

Reset VoIP Interface to Factory State—`voip_factory_reset`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
Event Source	No

Description

This parameter resets the VoIP plug-in card to its factory state.

to "disabled."

Forward a Call–`voip_forward`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

This parameter is used to forward a call.

Place Current Call on Hold—`voip_hold`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

Setting this parameter places the current call on hold.

Add Call to a Conference—`voip_join`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

This parameter is used to add a call to the conference.

Select the Active Line—`voip_line`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Sequence
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
System Limits	Minimum: 1, Maximum: 12
Event Source	No

Description

This parameter selects the currently active line. Some parameters (e.g. phone_connect, phone_dial, phone_redial) operate on the currently active line, as specified by this parameter. Setting this parameter is analogous to selecting a line key on the UI of a Polycom VoIP phone.

Label for the Line Key—`voip_line_label`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	String
Read/Write Mode	Read-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Indices	1-12 : Line number
System Limits	Max String Length: 128
Event Source	No

Description

This parameter reports the label for the specified line.

State for the VoIP Line—`voip_line_state`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Indices	1-12 : Line number

Argument	Argument value
Values	none messages do_not_disturb line_not_registered line_registered in_conference call_active call_on_hold shared_line speed_dial_indicator forward_all_calls acd_online acd_offline acd_not_logged_in acd_available remote_active secure_rtp remote_hold hd_audio offering proceed dial_tone held disconnect feat_enabled feat_disabled cma_presence_available cma_presence_busy cma_presence_available_in_a_call cma_presence_unavailable cma_presence_away cma_presence_offline ocs_available ocs_away ocs_busy ocs_do_not_disturb ocs_no_info ocs_offline blf_busy
Event Source	No

Description

This parameter reports the state for the specified line.

Reset Local Configuration Parameters—`voip_local_reset` I

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
Indices	1-12 : Line number
System Limits	Max String Length: 128
Event Source	No

Description

This parameter resets all local configuration parameters including the auto answer enable parameter.

Indicates Whether Messages are WaitingI—`voip_message_waiting`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read-Only
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
Event Source	Yes

Description

This parameter indicates whether or not a message is waiting for the VoIP interface on any of the registered lines.

Save VoIP Network Settings—`voip_net_cfg_save`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
Event Source	No

Description

This parameter causes the VoIP network settings to be written to the flash on the VoIP card.

Set Provisioning Server Address—`voip_prov_serv_address`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	String
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
System Limits	Max String Length : 256
Default	
Event Source	No

Description

This parameter sets the address of the provisioning server for the VoIP interface.

Set Provisioning Server Password—`voip_prov_serv_password`I

Argument	Argument value
Channel Type	Virtual Channel
Value Type	String
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
System Limits	Max String Length : 256
Default	
Event Source	No

Description

This parameter sets the password for the provisioning server used by the VoIP interface.

Set Provisioning Server Type—`voip_prov_serv_type` I

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
Values	ftp : FTP Server (default tftp : TFTP Server http : HTTP Server https : HTTPS Server ftps : FTPS Server
Event Source	No

Description

This parameter controls the provisioning server type for the VoIP interface.

Set Provisioning Server Username—`voip_prov_serv_user` I

Argument	Argument value
Channel Type	Virtual Channel
Value Type	String
Read/Write Mode	Read/Write
Phys Chans	VoIP Telephony Input
Virt Chans	Mono
Default	
Event Source	No

Description

This parameter sets the username for the provisioning server used by the VoIP interface.

Reboot VoIP Interface—`voip_reboot` I

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

Setting this parameter causes the SoundStructure VoIP Interface to reboot. The host SoundStructure device does not reboot when this parameter is used.

Resume a Call That is On Hold—`voip_resume` I

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

Setting this parameter takes the "active call" out of hold.

Send Call that can’t be Auto-dialed I—`voip_send`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

Setting this parameter causes a call to be placed with the digits dialed so far.

Add Call to Conference—`voip_split` I

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

This parameter splits all calls in a conference into individual calls on hold.

Status of the VoIP Interface—`voip_status` I

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Values	booting : The VoIP card is booting ok : The VoIP card has booted and is operational
Event Source	No

Description

This parameter indicates the status of the VoIP plug-in card. The three values correspond to the state of the status LED on the VoIP plug-in card as follows: ok = solid, booting = flashing.

Transfer a Call—`voip_transfer`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
Event Source	No

Description

This parameter is used to transfer a call.

Get the UC Software Version—`voip_uc_sw_ver`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	String
Read/Write Mode	Read-Only
Phys Chans	VoIP Telephony Output
Virt Chans	Mono
System Limits	Max String Length : 256
Event Source	No

Description

This parameter returns the UC software version of the VoIP plug-in card.

Equalizer Parameters

Description

The line outputs (both conferencing and sound reinforcement) each have a graphic equalizer that can have either 10 bands (1 octave), 15 bands (2/3 octave), or 31 bands (1/3 octave).

Most physical channel types have 10 bands of graphic equalization. The Conference Link input and output physical channel types only have 5 bands of graphic equalization. The signal generator and AEC reference physical channel types do not support graphic equalization.

All physical channel types except the signal generator support a high-pass filter, a low-pass filter, a high-shelf filter, and a low-shelf filter.

Enable All Equalizer Processing—eq_en

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) all equalizer processing (peq, geq, etc.) for the specified virtual channel.

Examples

Command	Response	Description
set eq_en "Amplifier" 1	val eq_en "Amplifier" 1	Enables the equalization processing for the channel "Amplifier".

Select Graphic or Parametric Equalizer—`eq_type`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output
Virt Chans	Mono, Stereo
Values	geq: Graphic Equalizer (default) peq: Parametric Equalizer
Event Source	No

Description

The line outputs may have either a graphic or parametric equalizer. This parameter selects which will be used for a given virtual channel.

Enable Gain Compensation For Graphic Equalizer—`geq_compensate`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) gain compensation for the graphic equalizer.

Enable Graphic Equalizer—`geq_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the graphic equalizer.

Gain of Graphic Equalizer Band—`geq_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output
Virt Chans	Mono, Stereo
Indices	1-31: Band number
System Limits	Minimum: -20.0, Maximum: 20.0, Resolution: 0.01
Default	0.0
User Limits Supported	No
Event Source	No

Description

Set the gain of the specified band in the graphic equalizer. The index must be between 1 and 10 for 1 octave equalization, between 1 and 15 for 2/3 octave equalization, and between 1 and 31 for 1/3 octave equalization.

Graphic Equalizer Type—`geq_type`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output
Virt Chans	Mono, Stereo
Values	1 : octave (10 band) 2/3 : 2/3 octave (15 band) 1/3 : 1/3 octave (31 band) (default)
Event Source	No

Description

This parameter sets the type of the graphic equalizer.

Enable High Shelving Filter—`high_shelf_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the high shelving filter for the specified virtual channel.

Frequency Of High Shelving Filter—`high_shelf_frequency`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum 20.0, Maximum: 20000.0, Resolution: 0.1
Default	500.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the cutoff frequency (in Hz) of the high shelving filter. This is the frequency at which the shelving filter's gain is half its maximum gain.

Gain Of High Shelving Filter—`high_shelf_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: -20.0, Maximum: 20.0, Resolution: 0.01
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the gain (in dB) of the high shelving filter at DC.

Slope Of High Shelving Filter—`high_shelf_slope`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Values	6: 6 dB per octave (default) 12: 12 dB per octave
Event Source	No

Description

This parameter sets the slope of the high shelving filter.

Enable Horn Equalizer—`horn_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the constant directivity horn equalizer for the specified virtual channel.

Frequency of Horn Equalizer—`horn_frequency`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Line Output, Sound Reinforcement Line Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 20.0, Maximum: 20000.0, Resolution: 0.1
Default	4000.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the cutoff frequency (in Hz) of the constant directivity horn equalizer. This is the frequency above which the gain increases at 6 dB per octave, and below which the gain is 0 dB.

Enable High-Pass Filter—`hpf_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the high-pass filter for the specified virtual channel.

Frequency Of High-Pass Filter—`hpf_frequency`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 20.0, Maximum: 20000.0, Resolution: 0.1
Default	160.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the frequency (in Hz) of the high-pass filter. For Butterworth filters this is the 3 dB frequency, but for Linkwitz-Riley filters, this is the 6 dB frequency.

Order of High-Pass Filter—`hpf_order`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 8
Default	2
User Limits Supported	No
Event Source	No

Description

This parameter sets the order of the high-pass filter. Linkwitz-Riley filters only support even orders. If an odd order is specified for a Linkwitz-Riley filter, it will be internally rounded up to an even number.

Type Of High-Pass Filter—`hpf_type`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Values	butterworth : Butterworth filter (default) linkwitz_riley : Linkwitz-Riley filter
Event Source	No

Description

This parameter sets the type of analog filter prototype used for the high-pass filter.

Enable Low Shelving Filter—`low_shelf_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the low shelving filter for the specified virtual channel.

Frequency Of Low Shelving Filter—`low_shelf_frequency`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 20.0, Maximum: 20000.0, Resolution: 0.1
Default	500.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the cutoff frequency (in Hz) of the low shelving filter. This is the frequency at which the shelving filter's gain is half its maximum gain.

Gain Of Low Shelving Filter—`low_shelf_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: -20.0, Maximum: 20.0, Resolution: 0.01
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the gain (in dB) of the low shelving filter at DC.

Slope Of Low Shelving Filter—`low_shelf_slope`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Values	6 : 6 dB per octave (default) 12 : 12 dB per octave
Event Source	No

Description

This parameter sets the slope of the low shelving filter.

Enable Low-Pass Filter—`lpf_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the low-pass filter for the specified virtual channel.

Frequency Of Low-Pass Filter—`lpf_frequency`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 20.0, Maximum: 20000.0, Resolution: 0.1
Default	16000.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the frequency (in Hz) of the low-pass filter. For Butterworth filters this is the 3 dB frequency, but for Linkwitz-Riley filters, this is the 6 dB frequency.

Order Of Low-Pass Filter—`lpf_order`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 8
Default	2
User Limits Supported	No
Event Source	No

Description

This parameter sets the order of the low-pass filter. Linkwitz-Riley filters only support even orders. If an odd order is specified for a Linkwitz-Riley filter, it will be internally rounded up to an even number.

Type Of Low-Pass Filter—`lpf_type`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Values	butterworth : Butterworth filter (default) linkwitz_riley : Linkwitz-Riley filter
Event Source	No

Description

This parameter sets the type of analog filter prototype used for the low-pass filter.

Enable Parametric Equalizer Band—`peq_band_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Indices	1-10 : Equalizer band
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the specified band of the parametric equalizer for the specified virtual channel. For conference link physical channels, the band index must be between 1 and 5. For all other physical channels, the band index must be between 1 and 10.

Bandwidth Of Parametric Equalizer Band—`peq_bandwidth`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Indices	1-10 : Equalizer band
System Limits	Minimum: 0.05, Maximum: 2.0, resolution: 0.01
Default	0.5
User Limits Supported	No
Event Source	No

Description

This parameter sets the bandwidth (in octaves) of the specified parametric equalizer band. In the case of peaking filters, this is the bandwidth at which the gain is half the peak gain (in dB). For notch filters, this is the 3 dB bandwidth. For all-pass filters, this is the bandwidth at which the phase shift is +/- 90 degrees. For conference link physical channels, the band index must be between 1 and 5. For all other physical channels, the band index must be between 1 and 10.

Frequency Of Parametric Equalizer Band—`peq_frequency`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Indices	1-10 : Equalizer band
System Limits	Minimum: 20.0, Maximum: 20000.0, Resolution: 0.1
Default	1000.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the frequency (in Hz) of the specified parametric equalizer band. In the case of peaking and notch types, this is the frequency at which the filter applies maximum (or minimum) gain. For all-pass filters, this is the frequency at which the phase shift is 180 degrees. For conference link physical channels, the band index must be between 1 and 5. For all other physical channels, the band index must be between 1 and 10.

Gain Of parametric Equalizer Band—`peq_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Indices	1-10 : Equalizer band
System Limits	Minimum: -20.0, Maximum: 20.0, Resolution: 0.01
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the gain (in dB) of the specified parametric equalizer band. For conference link physical channels, the band index must be between 1 and 5. For all other physical channels, the band index must be between 1 and 10.

Type Of parametric Equalizer Band—`peq_type`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Indices	1-10 : Equalizer band
Values	peq : peaking parametric equalizer (default) notch : notch filter allpass : 2nd order allpass
Event Source	No

Description

This parameter sets the type of the specified parametric equalizer band. For conference link physical channels, the band index must be between 1 and 5. For all other physical channels, the band index must be between 1 and 10.

Dynamics Processing Parameters

Description

Dynamics processing is available on all physical channels except the signal generator and AEC reference. Dynamics processing includes a compressor, limiter, expander, gate, and peak limiter. An additional input gain parameter is provided to change the gain of the signal before the dynamics processor.

Enable Dynamics Processing—`dp_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) all the dynamics processing for the specified virtual channel.

Gate Attack Time—`dp_gate_attack`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 200
Default	1
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of time (in milliseconds) it takes the dynamics gate to ramp the gain up to the target gain once the input signal level surpasses the gate threshold. This parameter does not affect the automixer processing.

Dynamics ‘Gate’ Decay Time—`dp_gate_decay`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 2000
Default	1000
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of time (in milliseconds) it takes the dynamics processing gate to ramp down to the target gain once the input signal drops below the gate threshold and the gate hold time has expired. This parameter does not affect the automixer processing.

Enable Gate—`dp_gate_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the gate function of the dynamics processor. This parameter and *dp_en* must be enabled for the gate to function. This parameter does not affect the automixer processing.

Gate Hold Time—`dp_gate_hold`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 2000
Default	500
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of time (in milliseconds) the input signal level must be below the dynamics gate threshold before the dynamics gate begins to decay. This parameter does not affect the automixer processing.

Gate Ratio—`dp_gate_ratio`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1.0, Maximum:100.0, Resolution: 0.1
Default	100.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the ratio of the target gain applied by the dynamics gate versus the difference between the input signal level and the gate threshold. For example, if the dynamics gate ratio is 10 (i.e., 10:1) and the input signal level is 6 dB below the gate threshold, the gate applies -60 dB of gain. This parameter does not affect the automixer processing.

Gate Threshold—`dp_gate_thresh`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: -100.0, Maximum: 0.0, Resolution: 0.1
Default	-100
User Limits Supported	No
Event Source	No

Description

This parameter sets the RMS level (in dBFS) of the input signal below which the dynamics gate engages. The level must be below this threshold longer than the gate hold time (set by *dp_gate_hold*) before the gate begins to apply a gain change. This parameter does not affect the automixer processing.

Expander Attack Time—`dp_exp_attack`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 200
Default	10
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of time (in milliseconds) it takes the expander to ramp the gain up to the target gain once the input signal level surpasses the expander threshold.

Expander Decay Time—`dp_exp_decay`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 2000
Default	100
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of time (in milliseconds) it takes the expander to ramp down to the target gain once the input signal drops below the expander threshold.

Enable Expander—`dp_exp_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the expander function of the dynamics processor. This parameter and *dp_en* must be enabled for the expander to function.

Expander Ratio—`dp_exp_ratio`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1.0, Maximum: 100.0, Resolution: 0.1
Default	2.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the ratio of the target gain applied by the expander versus the difference between the input signal level and the expander threshold. For example, if the expander ratio is 2 (i.e., 2:1) and the input signal level is 3 dB below the expander threshold, the expander applies -6 dB of gain.

Expander Threshold—`dp_exp_thresh`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: -100.0, Maximum: 0.0, Resolution: 0.1
Default	-100
User Limits Supported	No
Event Source	No

Description

This parameter sets the RMS level (in dBFS) of the input signal below which the expander engages.

Compressor Attack time—`dp_comp_attack`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 200
Default	10
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of time (in milliseconds) it takes the compressor to ramp the gain down to the target gain once the input signal level surpasses the compressor threshold.

Compressor Decay Time—`dp_comp_decay`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 2000
Default	100
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of time (in milliseconds) it takes the compressor to ramp the gain up to the target gain once the input signal level drops below the compressor threshold.

Enable Compressor—`dp_comp_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the compressor function of the dynamics processor. This parameter and *dp_en* must be enabled for the compressor to function.

Compressor Ratio—`dp_comp_ratio`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1.0, Maximum: 100.0, Resolution: 0.1
Default	2.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the ratio of the target gain applied by the compressor versus the difference between compressor threshold and the input signal level. For example, if the compressor ratio is 2 (i.e., 2:1) and the input signal level is 3 dB above the compressor threshold, the compressor applies -1.5 dB of gain.

Compressor Threshold—`dp_comp_thresh`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: -100.0, Maximum: 0.0, Resolution: 0.1
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the RMS level (in dBFS) of the input signal above which the compressor engages.

Limiter Attack Time—`dp_lim_attack`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 200
Default	5
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of time (in milliseconds) it takes the limiter to ramp the gain down to the target gain once the input signal level surpasses the limiter threshold.

Limiter Decay Time—`dp_lim_decay`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 2000
Default	500
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of time (in milliseconds) it takes the limiter to ramp the gain up to the target gain once the input signal level drops below the limiter threshold.

Enable Limiter—`dp_lim_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the limiter function of the dynamics processor. This parameter and *dp_en* must be enables for the limiter to function.

Limiter Ratio—`dp_lim_ratio`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 1.0, Maximum: 100.0, Resolution: 0.1
Default	10.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the ratio of the target gain applied by the limiter versus the difference between the limiter threshold and the input signal level. For example, if the limiter ratio is 10 (i.e., 10:1) and the input signal level is 6 dB above the limiter threshold, the limiter applies -5.4 dB of gain.

Limiter Threshold—`dp_lim_thresh`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: -100.0, Maximum: 0.0, Resolution: 0.1
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the RMS level (in dBFS) of the input signal above which the limiter engages.

Enable Peak Limiter—`dp_peak_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the peak limiter function of the dynamics processor. This parameter and *dp_en* must be enabled for the peak limiter to function.

Peak Limiter threshold—`dp_peak_thresh`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Input, Analog Telephony Output, Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: -100.0, Maximum: 0.0, Resolution: 0.1
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the RMS level (in dBFS) of the input signal above which the peak limiter engages. The peak limiter will ensure that the peak level never exceeds this threshold.

Algorithm Parameters

Enable Acoustic Echo Canceller—`aec_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the acoustic echo cancellation (AEC) algorithm.

Enable Noise Fill—`aec_noise_fill`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input
Virt Chans	Mono, Stereo
Default	1
Event Source	No

Description

This parameter enables (1) or disables (0) the noise fill algorithm in the AEC.

AEC Reference—`aec_ref`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	String
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input
Virt Chans	Mono, Stereo
Indices	1-2 : Left or right
System Limits	Max String Length: 256
Default	
Event Source	No

Description

This parameter is used to set the AEC references for an given virtual channel. The string argument specifies the name of the virtual channel that will be the AEC reference. The string argument must be a valid virtual channel name for a currently defined virtual channel that is a conferencing line output (`cr_line_out`), sound reinforcement line output (`sr_line_out`), or submix output (`submix`).

The index is used to specify the left (`1`) or right (`2`) reference channels. If neither the left nor the right channel have references specified, then the AEC is disabled. If only the left channel is specified, then the mono AEC algorithm is used. If both the left and right channel are specified, then the stereo AEC algorithm is used.

If the reference's virtual channel is mono, then the corresponding physical channel is used as the AEC reference. If the reference's virtual channel is stereo, then either the left or right physical channel is used as a reference, depending on which index is specified (`1` for left, `2` for right).

If the AEC is on a stereo virtual channel, then the reference specifications apply to both physical channels of the stereo virtual channel.

Enable Automatic Gain Control—`agc_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input, Analog Telephony Input
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the automatic gain control (AGC) algorithm.

AGC Maximum Gain—`agc_max_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input, Analog Telephony Input
Virt Chans	Mono, Stereo
System Limits	Minimum: 0.0, Maximum: 20.0, Resolution: 0.1
Default	6.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the maximum gain (in dB) that can be applied by the AGC.

AGC Minimum Gain—`agc_min_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input, Analog Telephony Input
Virt Chans	Mono, Stereo
System Limits	Minimum: -20.0, Maximum: 0.0, Resolution: 0.1
Default	-6.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the minimum gain (in dB) that can be applied by the AGC.

Amount Of Delay—`delay`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Output Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
System Limits	Minimum: 0, Maximum: 48000
Default	0
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of delay applied to the audio signal, in samples. The sampling frequency is 48 kHz, which means that a sample is 20.83 microseconds. The maximum delay of 96000 samples is equivalent to 2 seconds.

Enable Signal Delay—`delay_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Conferencing Line Output, Sound Reinforcement Mic/Line Input, Sound Reinforcement Line Output, Analog Telephony Output Submix, ConferenceLink Aux Input, ConferenceLink Aux Output
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (2) the delay algorithm.

Enable Feedback Reduction—`fb_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the feedback reduction algorithm.

Feedback Reduction filter Bandwidth—`fb_filter_bandwidth`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
System Limits	Minimum: 0.03, Maximum: 1.0, Resolution: 0.01
Default	0.1
User Limits Supported	No
Event Source	No

Description

This parameter sets the bandwidth (in octaves) for all the filters of the feedback reduction algorithm.

Enable Filter Decay Mode In Feedback Reduction Algorithm—`fb_filter_decay_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) filter decay mode for the feedback reduction algorithm. If filter decay mode is enabled, the adaptive notch filters can slowly decay to 0 dB if no singing is detected at that frequency. This mode is useful in rooms with high noise or where there is a lot of motion.

Reset One Of The Feedback Reduction Filters—`fb_filter_reset`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Void
Read/Write Mode	Write-Only
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Indices	1-10 : Filter number
Event Source	No

Description

Setting this parameter resets the specified filter in the feedback reduction algorithm. Redpoint will likely set this parameter for filters it has converted to fixed parametric EQ filters.

Maximum Filter Depth For Feedback Reduction Filters—`fb_filter_max_depth`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
System Limits	Minimum: -100.0, Maximum: 0.0, Resolution: 0.1
Default	-15.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the maximum attenuation (in dB) that can be applied for any feedback reduction filter.

Safe Mode Attenuation For Feedback Reduction—`fb_safe_mode_atten`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
System Limits	Minimum: 0.0, Maximum: 100.0, Resolution: 0.1
Default	3.0
User Limits Supported	No
Event Source	No

Description

This parameter defines the maximum amount of attenuation (in dB) applied to the input if all the filters are used up and the feedback reduction algorithm continues to detect singing. Setting this parameter to 0 dB means that no attenuation is performed even if all the filters are used up.

Select Mic audio Source Index—`mic_source_index`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Sequence
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Indices	1-32 : Physical channel
System Limits	Minimum: 1, Maximum: 15
Default	1
Event Source	No

Description

This parameter selects the index of the audio source for the corresponding `cr_mic_in` or `sr_mic_in` physical channels. The index required for this parameter indicates to which physical channel of the virtual channel this parameter will be applied. For example, indices `1` and `2` correspond to the left and right physical channels of a stereo virtual channel.

The value of this parameter indicates the index of the audio source type (*mic_source_type*) that will be routed to the physical channel.

When *mic_source_type* is `analog`, this parameter has no effect.

When *mic_source_type* is `clink_mic`, the value of this parameter indicates which ConferenceLink mic element will be routed to the physical channel.

For example, assume a virtual channel has been defined like this:

vcdef "Stereo Clink Mic" stereo cr_mic_in 1 2

And the source type has been set to `clink_mic` like this:

set mic_source_type "Stereo Clink Mic" clink_mic

Then we issue these commands:

set mic_source_index "Stereo Clink Mic" 1 5

set mic_source_index "Stereo Clink Mic" 2 6

These commands set the left and right channels of the "Stereo Clink Mic" virtual channel to use the 2nd and 3rd elements of the 2nd ConferenceLink mic.

Examples

Command	Response	Description
get mic_source_index 'Ceiling Mic 1 A'	val mic_source_index 'Ceiling Mic 1 A' 1	Queries which microphone index of the microphone array is associated with the virtual channel 'Ceiling Mic 1 A'. Since 'Ceiling Mic 1 A' is the first element of the digital array microphone, the value 1 is returned.
get mic_source_index 'Ceiling Mic 2 B'	val mic_source_index 'Ceiling Mic 2 B' 5	Queries which microphone index of the microphone array is associated with the virtual channel 'Ceiling Mic 2 B'.Since 'Ceiling Mic 2 B' is the second element of the second digital array microphone, the value 5 (=3+2) is returned.

Select Mic audio Source type—`mic_source_type`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Values	analog : Analog Mic Input (default) clink_mic : ConferenceLink Mic Input
Event Source	No

Description

This parameter selects the audio source for the corresponding cr_mic_in or sr_mic_in physical channels. The analog type selects the analog microphone audio. The clink_mic type selects one of the ConferenceLink microphone elements. Control of which element is selected is done through the *mic_source_index* parameter.

Examples

Command	Response	Description
set mic_source_type 'Table Mic 1' analog	val mic_source_type 'Table Mic 1' analog	Sets the mic_source_type for 'Table Mic 1' to analog.
get mic_source_type 'Ceiling Mic 1 A'	val mic_source_type 'Ceiling Mic 1 A' clink_mic	Queries the mic_source_type for 'Ceiling Mic 1 A'.

Enable Noise Canceller—`nc_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Analog Telephony Input
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables (1) or disables (0) the noise cancellation (NC) algorithm.

Noise Cancellation Level—`nc_level`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Analog Telephony Input
Virt Chans	Mono, Stereo
System Limits	Minimum: 0.0, Maximum: 20.0, Resolution: 1.0
Default	10.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of cancellation (in dB) applied by the noise cancellation algorithm.

Signal Generator Gain—`sig_gen_gain`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Signal Generator
Virt Chans	Mono
System Limits	Minimum: -100.0, Maximum: 20.0, Resolution: 0.1
Default	-30.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the gain of the signal produced by the signal generator, in dB. A gain of 0 dB will produce a signal that has the same RMS level as a sine wave at -20 dBFS (the nominal signal level). This means that not all signal types will have the same peak level, and some types may clip before a gain of 20 dB is applied.

Signal Generator Sweep Start Frequency—`sig_gen_sweep_start`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Signal Generator
Virt Chans	Mono
System Limits	Minimum: 20.0, Maximum: 20000.0, Resolution: 0.1
Default	20.0
User Limits Supported	No
Event Source	No

Description

When the signal generator’s *sig_gen_type* is set to sweep, this parameter sets the frequency (in Hz) at which the sweep generator begins. The direction of the frequency sweep will be up or down depending on whether this parameter is higher or lower than the *sig_gen_sweep_stop* parameter.

Signal Generator Sweep Step Size—`sig_gen_sweep_step`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Signal Generator
Virt Chans	Mono
Values	continuous : Continuous (default) 1/24 : 1/24 Octave 1/12 : 1/12 Octave 1/6 : 1/6 Octave 1/3 : 1/3 Octave 1 : 1 Octave
Event Source	No

Description

When the signal generator's *sig_gen_type* is set to sweep, this parameter sets the step size of the sweep generator. This signal generator can sweep continuously, in fractional octave steps, or in full octave steps.

Signal Generator Sweep Stop Frequency—`sig_gen_sweep_stop`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Signal Generator
Virt Chans	Mono
System Limits	Minimum: 20.0, Maximum: 20000.0, Resolution: 0.1
Default	20000.0
User Limits Supported	No
Event Source	No

Description

When the signal generator's *sig_gen_type* is set to sweep, the parameter sets the frequency (in Hz) at which the sweep generator stops. The direction of the frequency sweep will be up or down depending on whether the *sig_gen_sweep_start* parameter is higher or lower than this parameter.

Signal Generator Sweep Time—`sig_gen_sweep_time`

Argument	**Argument value**
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Signal Generator
Virt Chans	Mono
System Limits	Minimum: 10, Maximum: 60000
Default	10000
User Limits Supported	No
Event Source	No

Description

When the signal generator's *sig_gen_type* is set to sweep, this parameter sets the duration (in milliseconds) that the sweep generator takes to sweep from its start frequency to its stop frequency.

Signal Generator Tone Frequency—`sig_gen_tone_freq`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Signal Generator
Virt Chans	Mono
System Limits	Minimum: 20.0, Maximum: 20000.0, Resolution: 0.1
Default	1000.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the frequency (in Hz) of the sine wave produced by the signal generator when its *sig_gen_type* is set to tone.

Signal Generator Type—`sig_gen_type`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Signal Generator
Virt Chans	Mono
Values	pink : Pink Noise (default) white : White Noise tone : Sine Wave sweep : Sine Wave Sweep
Event Source	No

Description

This parameter sets the type of signal produced by the signal generator. The options are pink noise (pink), white noise (white) a sine wave at a single frequency (tone) and a sine wave swept across a range of frequencies (sweep).

Input Path Parameters

Select Processing For Ungated Signal—`cr_ungated_type`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input
Virt Chans	Mono, Stereo
Values	conf : Conferencing sr : Sound Reinforcement line : Line Input (default) bypass : Bypass
Event Source	No

Description

This parameter selects the version of signal to use for the ungated triune signal of the specified virtual channel.

elect Delay for Sound Reinforcement Signal—`sr_delay_type` S

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Values	normal : Normal (default) low_delay : Low Delay
Event Source	No

Description

This parameter selects the version of signal to use for the sound reinforcement triune signal of the specified virtual channel.

Select Processing For Ungated Signal—`sr_ungated_type`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	List
Read/Write Mode	Read/Write
Phys Chans	Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Values	rec : Recording sr : Sound Reinforcement line : Line Input (default) bypass : Bypass
Event Source	No

Description

This parameter selects the version of signal to use for the ungated triune signal of the specified virtual channel.

Enable Delay Compensation For Triune Signals—`ungated_delay_comp_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

Delay compensation only applies to the ungated signal, and only when it is using the line or bypass options.

Automixer Parameters

Automixer Adaptive Threshold—`am_adapt_thresh`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
System Limits	Minimum: 0.0, Maximum: 100.0, Resolution: 0.1
Default	10.0
User Limits Supported	No
Event Source	No

Description

This parameter defines how much louder (in dB) the microphone's signal level must be above its measured noise floor before it is eligible to be considered active. Higher settings will make the microphone's gating less sensitive, while lower settings will make it more sensitive.

Automixer Camera Activity Time—`am_camera_activity_time`

Argument	Argument value
Channel Type	Global System
Value Type	Integer
Read/Write Mode	Read/Write
Indices	1-63 : group number
System Limits	Minimum: 10, Maximum: 10000
Default	2000
User Limits Supported	No
Event Source	No

Description

This parameter defines the amount of time (in ms) a signal must be active before showing up on the camera activity meter. In general, it should be set somewhat longer than the hold time of the automixer.

Automixer Chairman Microphone—`am_chairman`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

When this parameter is set to 1, the microphone is considered a chairman microphone.

Automixer Channel Bias—`am_chan_bias`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Floating-Point
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
System Limits	Minimum: -20.0, Maximum: 20.0, Resolution: 0.1
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the channel bias (in dB) for the associated microphone. For the purpose of determining activity status, this microphone is treated as though its level were higher or lower (according to the setting of the parameter) than its actual measured level.

Automixer Decay Time—`am_decay_time`

Argument	Argument value
Channel Type	Global System
Value Type	Integer
Read/Write Mode	Read/Write
Indices	1-63 : group number
Systems Limits	Minimum: 10, Maximum: 10000
Default	1000
User Limits Supported	No
Event Source	No

Description

This parameter defines how long (in ms) the gain of a gated microphone in the specified automixer group takes to transition between fully open and its off attenuation value when it is time for the microphone to gate off.

Enable Automixer—`am_en`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

This parameter enables or disables the automixer for the virtual channel. When disabled, the microphone is completely pulled out of the automixer (so that it does not affect the gating of other channels), and a gain of 0 dB is applied to the channel (so that it is always open).

Examples

Command	Response	Description
set am_en "Table Mic 1" 1	val am_en "Table Mic 1" 1	Enables the automixer processing for the channel "Table Mic 1".

Enable Gain-Sharing Automixer Mode—`am_gain_sharing`

Argument	Argument value
Channel Type	Global System
Value Type	Boolean
Read/Write Mode	Read/Write
Indices	1-63 : group number
Default	0
Event Source	No

Description

This parameter selects gain-sharing mode for the specified automixer group when set to 1. Otherwise, the microphones in the automixer group are in gating mode.

Examples

Command	Response	Description
set am_gain_sharing 2 1	val am_gain_sharing 2 1	Selects the gain sharing automixer for the microphones in automixer group 2.
set am_gain_sharing 1 0	val am_gain_sharing 1 0	Selects the gated automixer for the microphones in automixer group 1.

Automixer Group—`am_group`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Sequence
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Systems Limits	Minimum: 1, Maximum: 63
Default	1
Event Source	No

Description

This parameter selects the automixer group in which the microphone is a member.

Examples

Command	Response	Description
set am_group "Table Mic 1" 1	val am_group "Table Mic 1" 1	Assigns "Table Mic 1" to automixer group 1.

Automixer Hold Time—`am_hold_time`

Argument	Argument value
Channel Type	Global Systems
Value Type	Integer
Read/Write Mode	Read/Write
Indices	1-63 : group number
Systems Limits	Minimum: 100, Maximum: 10000
Default	500
User Limits Supported	No
Event Source	No

Description

This parameter defines how long (in ms) the microphone in the specified automixer group will be considered active after the last detected significant level on the microphone.

Automixer Last Mic Mode—`am_last_mic_mode`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
Default	0
Event Source	No

Description

When this parameter is set to 1, the last mic mode is enabled on the microphone.

NOM Limit—`am_nom_limit`

Argument	Argument value
Channel Type	Global System
Value Type	Integer
Read/Write Mode	Read/Write
Indices	1-63 : group number
System Limits	Minimum: 0, Maximum: 128
Default	16
User Limits Supported	No
Event Source	No

Description

This parameter sets the NOM limit for the microphone with respect to its automixer group.

Automixer Off Attenuation—`am_off_atten`

Argument	Argument value
Channel Type	Global System
Value Type	Floating-Point
Read/Write Mode	Read/Write
Indices	1-63 : group number
System Limits	Minimum: 0.0, Maximum: 100.0, Resolution: 0.1
Default	15.0
User Limits Supported	No
Event Source	No

Description

This parameter defines how much attenuation (in dB) is applied to a gated microphone in the specified group when the microphone is fully gated off.

This parameter is only used if the automixer group is in gating mode. If it is gain sharing mode, the parameter is ignored.

Automixer Microphone Priority—`am_priority`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Conferencing Mic/Line Input, Sound Reinforcement Mic/Line Input
Virt Chans	Mono, Stereo
System Limits	Minimum: 1, Maximum: 4
Default	1
User Limits Supported	No
Event Source	No

Description

This parameter sets the priority of the microphone. A priority of 1 is the highest priority (most favored), while a priority of 4 is the lowest priority (least favored).

Automixer Priority Attenuation—`am_priority_atten`

Argument	Argument value
Channel Type	Global System
Value Type	Floating-Point
Read/Write Mode	Read/Write
Indices	1-63 : group number
System Limits	Minimum: 0.0, Maximum: 100.0, Resolution: 0.1
Default	0.0
User Limits Supported	No
Event Source	No

Description

This parameter sets the amount of attenuation (in dB) that is applied to the microphones in the specified automixer group if a higher priority microphone in the group is currently active.

Gain Sharing Automixer Slope—`am_slope`

Argument	Argument value
Channel Type	Global System
Value Type	Floating-Point
Read/Write Mode	Read/Write
Indices	1-63 : group number
System Limits	Minimum: 0.0, Maximum: 10.0, Resolution: 0.1
Default	2.0
User Limits Supported	No
Event Source	No

Description

This parameter defines how much attenuation (in dB) is applied to microphones in the specified automixer group when they don't have the highest level in the group. For example, if a microphone has a level that is 6.0 dB lower than the loudest mic, and its slope is 2.0, then 12.0 dB of attenuation will be applied to the microphone.

This parameter is only used if the automixer group is in gain sharing mode. If the automixer group is in gating mode, the parameter is ignored.

GPIO Control Parameters

Analog GPIO Value—`analog_gpio_value`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Analog General Purpose I/O Input
Virt Chans	Control
System Limits	Minimum: 0, Maximum: 255
Default	0
User Limits Supported	No
Event Source	Yes

Description

This parameter gets or sets the value of the analog gpio pin. Writing an input has no effect and returns the current value of the input. The value for this parameter is an integer between *analog_gpio_min* and *analog_gpio_max*, inclusive. Those parameters control how the analog value of the pin is mapped to an integer range.

Examples

Command	Response	Description
get analog_gpio_value "Analog Logic Pin"	val analog_gpio_value "Analog Logic Pin" 0	Returns the analog voltage associated with the analog logic pin "Analog Logic Pin". The logic pin "Analog Logic Pin" must have been created with a vcdef command.

Digital GPIO Pin held Status—`digital_gpio_held`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read-Only
Phys Chans	Digital General Purpose I/O Input
Virt Chans	Control
Event Source	Yes

Description

This parameter indicates when a digital input pin is held. It is similar in functionality to the *ir_key_held* parameter. The hold time and repeat time are specified by the *digital_gpio_hold_time* and *digital_gpio_repeat_time* parameters. When the pin is held for the hold time, a status message will be generated. If the pin remains held, status messages will be generated with a period equal to the repeat time.

Digital GPIO Pin Hold Time—`digital_gpio_hold_time`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Digital General Purpose I/O Input
Virt Chans	Control
System Limits	Minimum: 0, Maximum: 10000
Default	0
User Limits Supported	No
Event Source	No

Description

This parameter specifies the amount of time (in milliseconds) that a GPIO pin must be held for the first *digital_gpio_held* status message to be sent. Setting this parameter to 0 indicates that *digital_gpio_held* messages will not be generated.

Digital GPIO Pin Repeat Time—`digital_gpio_repeat_time`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Digital General Purpose I/O Input
Virt Chans	Control
System Limits	Minimum: 1, Maximum: 10000
Default	1000
User Limits Supported	No
Event Source	No

Description

This parameter specifies the amount of time (in milliseconds) between *digital_gpio_held* status messages when a GPIO pin is continually held.

Digital GPIO Pin Status—`digital_gpio_state`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Boolean
Read/Write Mode	Read/Write
Phys Chans	Digital General Purpose I/O Input, Digital General Purpose I/O Output
Virt Chans	Control
Default	0
Event Source	Yes

Description

This parameter gets or sets the value of the digital gpio pin. Writing an input has no effect and returns the current value of the input.

Digital GPIO Array Value—`digital_gpio_value`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Digital General Purpose I/O Input, Digital General Purpose I/O Output
Virt Chans	Control Array
System Limits	Minimum: 0, Maximum: 2147483647
Default	0
User Limits Supported	No
Event Source	Yes

Description

This parameter gets or sets the value of the digital gpio array. Writing an input has no effect and returns the current value of the input.

Control Port Parameters

Set Authentication Password—`auth_password`

Argument	Argument value
Channel Type	Global system
Value Type	String
Read/Write Mode	Write-Only
System Limits	Max String Length : 128
Event Source	No

Description

This parameter sets the authentication password. The default value for this parameter is “456”. The settings are permanently changed immediately after the command has been executed.

Examples

Command	Response	Description
set auth_password “12345”	val auth_password “****”	Sets the authentication password to “12345”. The acknowledgment masks the password.

Clink2 Call Active Status—`clink_call_active`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	Boolean
Read/Write Mode	Read-Only
Event Source	Yes

Description

This parameter gets the call active status of a Polycom Video Codec. This is true whenever the Polycom Video Codec has an active video or PSTN call. A status message is generated whenever the call active status is changed by a Polycom Video Codec.

Clink2 Local Call Active Status—`clink_local_call_active`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	Integer
Read/Write Mode	Read/Write
System Limits	Minimum: 0, Maximum: 32
Default	0
User Limits Supported	No
Event Source	Yes

Description

This parameter gets and sets the call active status that the device broadcasts to any connected Polycom Video Codec and Polycom Microphone Arrays. This primarily controls the state of the green LED on the Polycom Microphone Arrays. Whenever this parameter is a value greater than 0, the call active status sent to the Polycom Video Codec and Polycom Microphone Arrays is set to true. This is implemented as an integer command so that when a call of interest goes active, it can be incremented. When the call goes inactive, this parameter can be decremented. This provides a count of all active calls in the system.

Clink2 Mute Status—`clink_mute`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	Boolean
Read/Write Mode	Read/Write
Default	0
Event Source	Yes

Description

This parameter gets or sets the mute status of a Polycom Video Codec or Polycom Microphone Array attached to the indicated device. This does not actually mute any audio. It only reflects the settings of the red mute LEDs on Polycom Microphone Arrays, or the mic mute display on a Polycom Video Codec. A status message is generated whenever the mute status is changed by a Polycom Video Codec or Polycom Microphone Array.

CLink2 Volume Status—`clink_volume`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	Integer
Read/Write Mode	Read/Write
System Limits	Minimum: 0, maximum: 51
Default	31
User Limits Supported	No
Event Source	Yes

Description

This parameter gets or sets the volume status of a Polycom Video Codec. This does not actually adjust any gains. It only reflects the settings of the on-screen volume control bar of the Polycom Video Codec. A status message is generated whenever the volume is changed by a Polycom Video Codec.

Ethernet Authentication Mode—`eth_auth_mode`

Argument	Argument value
Channel Type	Global System
Value Type	List
Read/Write Mode	Read/Write
Values	open : Unauthenticated connections on port 52774 (default) auth : Authenticated connections on port 52775
Event Source	No

Description

This parameter selects the port for Ethernet control connections. If set to open, then connections are accepted on port 52774 and do not require authentication. If it is set to auth, then connections are accepted on port 52775 and authentication is required.

Ethernet Settings—`eth_settings`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	String
Read/Write Mode	Read/Write
System Limits	Max String Length: 256
Default	mode=’dhcp’
Event Source	No

Description

This parameter gets or sets the Ethernet settings. When this parameter is set, the Ethernet interface is automatically restarted with the new settings.

The format of the string is a comma-separated list of name/value pairs with the name and value separated by an equals character, and the value enclosed in single quotes.

The `mode` attribute is always required. It must be either `dhcp` or `static`.

The `addr`, `dns`, `gw`, and `nm` attributes are required when `mode` is `static` and ignored (not required) when `mode` is `dhcp`. They are always returned in the acknowledgment. For the `dhcp` case, they specify the values obtained from the DHCP server.

The `addr` attribute specifies the IP address of the interface. The `dns` attribute specifies the domain name server(s). A single server or multiple servers (separated by spaces) may be specified. The `gw` attribute specifies the gateway. The `nm` parameter specifies the netmask.

DHCP Example

set eth_settings 1 "mode='dhcp'"

val eth_settings 1 "mode='dhcp',addr='172.22.2.129',dns='172.22.1.1 172.22.1.2',gw='172.22.2.254',nm='255.255.255.0'"

Static IP Example

set eth_settings 1 "mode='static',addr='172.22.2.200',dns='172.22.1.1',gw='172.22.2.254',nm='255.255.255.0'"

val eth_settings 1 "mode='static',addr='172.22.2.200',dns='172.22.1.1',gw='172.22.2.254',nm='255.255.255.0'"

Examples

Command	Response	Description
set eth_settings 1 "mode='dhcp'"	val eth_settings 1 "mode='dhcp',addr='172.22.2.129',dns='172.22.1.1 172.22.1.2',gw='172.22.2.254',nm='255.255.255.0'"	Sets the Ethernet settings on device 1 to dhcp and returns the full Ethernet settings.
set eth_settings 1 "mode='static',addr='192.168.10.63',dns='192.168.10.1',gw='192.168.10.254',nm='255.255.255.0'"	val eth_settings 1 "mode='static',addr='192.168.10.63',dns='192.168.10.1',gw='192.168.10.254',nm='255.255.255.0'"	Sets the Ethernet settings on device 1 to the static IP address of 192.168.10.63 and returns the full Ethernet settings. All parameters must be specified even if just changing one parameter.
get eth_settings 1	val eth_settings 1 "mode='static',addr='192.168.10.63',dns='192.168.10.1',gw='192.168.10.254',nm='255.255.255.0'"	Queries the Ethernet settings on device 1.

Get Ethernet MAC Address—`eth_mac`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	String
Read/Write Mode	Read-Only
System Limits	Max String Length: 17
Event Source	No

Description

This parameter gets the MAC address for the system's Ethernet port. The value will be formatted in all lowercase with bytes separated by a colon, for example “00:04:f2:bf:00:01”.

Examples

Command	Response	Description
get eth_mac 1	val eth_mac 1 "00:04:f2:bf:00:01"	Queries the Ethernet mac address on device 1.

Key Pressed On IR Remote—`ir_key_press`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read-Only
Phys Chans	Infrared Remote Input
Virt Chans	Control
System Limits	Minimum: 0, Maximum: 255
Event Source	No

Description

When queried, this parameter returns the keycode value of the last key that was pressed on the IR remote. As an event, a status message is generated whenever a key is pressed on the IR remote.

Key Held On IR Remote—`ir_key_held`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Infrared Remote Input
Virt Chans	Control
System Limits	Minimum: 0, Maximum: 255
Event Source	No

Description

When queried, this parameter returns the keycode value of the last key that was held on the IR remote. As an event, a status message is generated at an interval of approximately 100 ms whenever an key is held on the IR remote.

Set IR Remote Channel ID—`ir_chan_id`

Argument	Argument value
Channel Type	Virtual Channel
Value Type	Integer
Read/Write Mode	Read/Write
Phys Chans	Infrared Remote Input
Virt Chans	Control
System Limits	Minimum: 0, Maximum: 15
Default	3
User limits Supported	No
Event Source	No

Description

This parameter sets the channel ID that the specified IR input will respond to.

Polycom IR remotes can be configured to use different channel IDs so that multiple remotes can be used in the same room to control different equipment without interfering with each other. By default, the Polycom IR remote used channel ID 3. This can be changed by following the instructions in the Administrator's Guide for Polycom HDX System's.

RS-232 Baud Rate—`ser_baud`

Argument	**Argument value**
Channel Type	Device-Specific System
Value Type	List
Read/Write Mode	Read/Write
Values	9600 : 9600 bits per second (default) 19200 : 19200 bits per second 38400 : 38400 bits per second 57600 : 57600 bits per second 115200 : 115200 bits per second
Event Source	No

Description

This parameter sets the baud rate for the RS-232 port. Hardware flow control should be enabled for baud rates over 9600 bps (see the *ser_flow* parameter).

Examples

Command	Response	Description
set ser_baud 1 9600	val ser_baud 1 9600	Sets the serial port baud rate on device 1 (the first device) to 9600.

Set RS-232 Control Mode—`ser_control_mode`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	List
Read/Write Mode	Read/Write
Values	command : Command mode (default) broadcast : Broadcast mode
Event Source	No

Description

This parameter sets the mode of operation for the RS-232 port. When set to *command*, the RS-232 port is operating as an interface to the SoundStructure command processor. When set to *broadcast* the *ser_send* parameter can be used to send arbitrary commands to control other devices connected to the RS-232 port. In *broadcast* mode, all received data is ignored.

RS-232 Flow Control—`ser_flow`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	List
Read/Write Mode	Read/Write
Values	none : No flow control (default) hw : Hardware flow control (RTS/CTS)
Event Source	No

Description

This parameter sets the type of flow control that will be used on the RS-232 port. Hardware flow control is recommended for baud rates over 9600 bps.

Examples

Command	Response	Description
set ser_flow 1 hw	val ser_flow 1 hw	Sets the serial port flow control on device 1 (the first device) to 'hw'.
set ser_flow 1 none	val ser_flow 1 none	Disables the serial port flow control on device 1 (the first device) by setting the flow control to 'none'.

Send Arbitrary Data to RS-232 Port—`ser_send`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	String
Read/Write Mode	Read-Only
System Limits	Max String Length: 256
Event Source	No

Description

This parameter is used to broadcast arbitrary commands to equipment attached to the RS-232 port. If the ser_control_mode parameter is set to *broadcast* for the RS-232 port, then the data in the string will be sent to the RS-232 port. If it is set to *command* then setting this parameter has no effect.

Special characters may be included in the string by escaping them. Since the string format already uses a backslash to escape double quote characters, a double backslash must be used to escape the special characters for this parameter. The following escape sequences are supported.

- \\ -- a single backslash character
- \\n -- new line
- \\r -- carriage return
- \\xNN -- byte value in hexadecimal (must contain exactly two digits)

Examples

Command	Response	Description
set ser_send 1 “Hello, World\\r”	set ser_send 1 “Hello, World\\r”	Send the string “Hello, World” followed by a carriage return to the RS-232 port on device 1.
set ser_send 1 “\\x48\\x65\\x6c\\x6c\\x6f\\x2c\\x20\\x57\\x6f\\x72\\x6c\\x64\\x0d”	set ser_send 1 \\x48\\x65\\x6c\\x6c\\6f\\x2c\\x20\\x57\\x6f\\x72\\x6c\\x64\\x0d”	Send the same string as the previous example, but using hexadecimal to specify the bytes instead.

System Parameters

Bootloader Version—`dev_bootloader_ver`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	String
Read/Write Mode	Read-Only
System Limits	Max String Length: 24
Event Source	No

Description

This parameter returns the bootloader version.

Firmware Version—`dev_firmware_ver`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	String
Read/Write Mode	Read-Only
System Limits	Max String Length: 24
Event Source	No

Description

This parameter returns the device’s firmware version.

Examples

Command	Response	Description
get dev_firmware_ver 1	val dev_firmware_ver 1 "1.0.0"	Returns the revision of the firmware for device 1 (the first device).

Hardware ECO Number—`dev_hw_eco`

Argument	**Argument value**
Channel Type	Device-Specific System
Value Type	Integer
Read/Write Mode	Read-Only
System Limits	Minimum: 0, Maximum: 255
Event Source	No

Description

This parameter returns the hardware ECO number for the board. The number does not track the actual ECO number, but rather indicates major ECO changes that we may need to account for in software.

Hardware Revision—`dev_hw_rev`

Argument	**Argument value**
Channel Type	Device-Specific System
Value Type	String
Read/Write Mode	Read-Only
System Limits	Max String Length: 32
Event Source	No

Description

This parameter returns the hardware revision of the device. Typical values are “X1”, “X2”, “X3”, “4”, etc.

Cycle Front Panel LED—`dev_led_cycle`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	Integer
Read/Write Mode	Write-Only
System Limits	Minimum: 1, Maximum: 60
Event Source	No

Description

This parameter causes the front panel LED of the specified device to cycle through its colors (yellow-red-green-off) for the specified number of seconds.

NTP Server—`dev_ntp_server`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	String
Read/Write Mode	Read/Write
System Limits	Max String Length: 32
Default	pool.ntp.org
Event Source	No

Description

This parameter gets or sets the name of the network time protocol (NTP) server used to set the system time. A power cycle is required for this parameter to take effect.

System Status—`dev_status`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	List
Read/Write Mode	Read-Only
Values	ok : Normal operation warning : Warning condition error : Error condition
Event Source	No

Description

This parameter returns the status of the system. A value of ok indicates that the system is operating normally. The front-panel LEDs on all the devices will be green in this condition. A value of warning indicates that a warning condition has occurred. A warning condition is usually due to a configuration error that can be corrected via software. The front-panel LEDs on all of the devices will be yellow in this condition. A value of error indicated that an error has occurred that is most likely due to a hardware failure or some other serious condition that can't be corrected via software. The front-panel LEDs on one or more of the linked devices will be red in this condition.

Internal Temperature—`dev_temp`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	Floating-Point
Read/Write Mode	Read-Only
Indices	1-3 : Temperature sensor index
System Limits	Minimum: -40.0, Maximum: 125.0, Resolution: 0.1
Event Source	No

Description

This parameter returns the temperature (in degrees C) from of one of the internal temperature sensors. The temperature sensors have the following locations by index. 1 is at the back right, underneath the plug-in slot. 2 is near the center of the analog input circuitry. 3 is at the front right, in front of the power supply.

Internal Temperature Status—`dev_temp_status`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	List
Read/Write Mode	Read-Only
Values	ok : Normal operation warning : Warning condition error : Error condition
Event Source	Yes

Description

this parameter returns the temperature status of the specified device. If the internal temperature sensors indicated the device is operating within a safe temperature rage, the *ok* value is returned. The *warning* value is returned when the temperature is in the marginal temperature range. The *error* value is returned when the temperature is in an unsafe temperature range.

Device Type—`dev_type`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	List
Read/Write Mode	Read-Only
Values	c16 : Conferencing 16x16 c12 : Conferencing 12x12 c8 : Conferencing 8x8 sr16 : Sound Reinforcement 16x16 sr12 : Sound Reinforcement 12x12 sr8 : Sound Reinforcement 8x8
Event Source	No

Description

This parameter returns the type of the device.

System Uptime—`dev_uptime`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	String
Read/Write Mode	Read-Only
System Limits	Max String Length: 16
Event Source	No

Description

This parameter returns the amount of time since the last reboot. The value returned is formatted as days:hours:minutes:seconds. For example, a value of “247:02:14:31” indicates the system has been running for 247 days, 2 hours, 14 minutes, and 31 seconds.

ConferenceLink Supply Voltage—`dev_volt_clink`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	Floating-Point
Read/Write Mode	Read-Only
System Limits	Minimum: 0.0, Maximum: 50.5, Resolution: 0.1
Event Source	No

Description

This parameter returns the voltage (in Volts) of the ConferenceLink power supply.

-15 V Supply Voltage—`dev_volt_neg_15`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	Floating-Point
Read/Write Mode	Read-Only
System Limits	Minimum: -14.8, Maximum: 0.0, Resolution: 0.1
Event Source	No

Description

This parameter returns the voltage (in Volts) of the -15 V power supply.

Phantom Power Supply Voltage—`dev_volt_phantom`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	Floating-Point
Read/Write Mode	Read-Only
Indices	1-4 : Phantom power bank index
System Limits	Minimum: 0.0, Maximum: 50.5, Resolution: 0.1
Event Source	No

Description

This parameter returns the output voltage (in Volts) of the specified phantom power supply. There is one phantom power supply for each bank of four inputs. Thus, there are two banks on an 8x8, three banks on a 12x12, and four banks on a 16x16. The voltage will be 0 for supplies that are not present on a device. Index 1 corresponds to the phantom power supply on mic inputs 1-4, index 2 corresponds to the phantom power supply on mic inputs 5-8, and so on.

+15 V Supply Voltage—`dev_volt_pos_15`

Argument	Argument value
Channel Type	Device-Specific System
Value Type	Floating-Point
Read/Write Mode	Read-Only
System Limits	Minimum: 0.0, Maximum: 14.8, Resolution: 0.1
Event Source	No

Description

This parameter returns the voltage (in Volts) of the +15 V power supply.

Enable/disable reporting of cmd/ack in logs—`sys_cmd_log_en`

Argument	Argument value
Channel Type	Global System
Value Type	Boolean
Read/Write Mode	Read/Write
Default	1
Event Source	No

Description

This parameter controls whether commands and acknowledgments are reported in the system logs.

Restore System To Factory Settings—`sys_factory_reset`

Argument	Argument value
Channel Type	Global System
Value Type	Void
Read/Write Mode	Write-Only
Event Source	No

Description

Setting this parameter restores the device to its factory settings, erasing all user data but retaining the current version of firmware.

A sys_factory_reset can also be performed by powering up the SoundStructure device with RS-232 pins 8 and 9 shorted together.

Get Last Executed Full Preset—`sys_last_full_preset`

Argument	Argument value
Channel Type	Global System
Value Type	String
Read/Write Mode	Read-Only
Event Source	No

Description

This parameter returns the name of the last executed full preset.

Get Last Executed Partial Preset—`sys_last_partial_preset`

Argument	Argument value
Channel Type	Global System
Value Type	String
Read/Write Mode	Read-Only
Event Source	No

Description

This parameter returns the name of the last executed partial preset.

Get Last Executed Preset—`sys_last_preset`

Argument	Argument value
Channel Type	Global System
Value Type	String
Read/Write Mode	Read-Only
Event Source	No

Description

This parameter returns the name of the last executed preset (either partial or full preset).

Enable/Disable Reporting Of mtrreg/mtrunreg In Logs—`sys_mtrreg_log_en`

Argument	Argument value
Channel Type	Global System
Value Type	Boolean
Read/Write Mode	Read/Write
Default	1
Event Source	No

Description

This parameter controls whether mtrreg and mtrunreg commands and acknowledgments are reported in the system logs.

System Name—`sys_name`

Argument	Argument value
Channel Type	Global System
Value Type	String
Read/Write Mode	Read/Write
System Limits	Max String Length: 256
Default	SoundStructure System
Event Source	No

Description

This parameter sets the name of the system.

Number of auth Ethernet connections—`sys_num_auth_connections`

Argument	Argument value
Channel Type	Global System
Value Type	Integer
Read/Write Mode	Read-only
System Limits	Minimum: 0, Maximum: 256
Event Source	No

Description

This parameter returns the total number of Ethernet connections to the system for which the parameter eth_auth_mode is set to auth.

Number of Ethernet connections—`sys_num_connections`

Argument	Argument value
Channel Type	Global System
Value Type	Integer
Read/Write Mode	Read-only
System Limits	Minimum: 0, Maximum: 256
Event Source	No

Description

This parameter returns the total number of Ethernet connections to the system.

Number of Devices—`sys_num_devs`

Argument	Argument value
Channel Type	Global System
Value Type	Integer
Read/Write Mode	Read-only
System Limits	Minimum: 1, Maximum: 8
Event Source	No

Description

This parameter returns the total number of devices connected over the OBAM bus.

Number of open Ethernet connections—`sys_num_open_connections`

Argument	Argument value
Channel Type	Global System
Value Type	Integer
Read/Write Mode	Read-only
System Limits	Minimum: 0, Maximum: 256
Event Source	No

Description

This parameter returns the total number of Ethernet connections to the system for which eth_auth_mode is set to open.

Pause System Execution—`sys_pause`

Argument	Argument value
Channel Type	Global System
Value Type	Integer
Read/Write Mode	Write-Only
System Limits	Minimum: 1, Maximum: 5000
Event Source	No

Description

This parameter pauses system execution for the specified number of milliseconds. Note that the entire command processor is paused, affecting all communication ports.

The typical application for this parameter is inserting pauses between commands in partial preset execution.

Plug-in Cards Match Configuration—`sys_plugins_match`

Argument	Argument value
Channel Type	Global System
Value Type	Boolean
Read/Write Mode	Read-only
Event Source	No

Description

This parameter returns true (1) if the actual plug-in cards in the system match the stored configuration.

Reset The Device—`sys_reboot`

Argument	Argument value
Channel Type	Global System
Value Type	Void
Read/Write Mode	Write-Only
Event Source	No

Description

Setting this parameter causes all linked devices to reboot as if a power-cycle has occurred.

Appendix B: Address Book

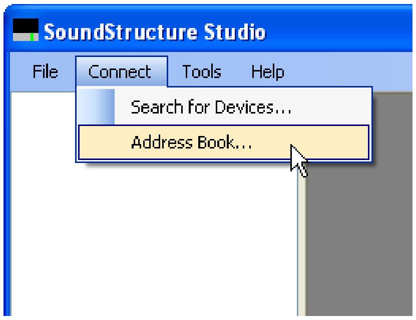
This chapter describes the SoundStructure Studio address book that can be used to organize and store IP addresses of SoundStructure systems. For SoundStructure systems that are not on the same subnet as the computer running SoundStructure Studio, the address book is a convenient way to store information about SoundStructure systems and provides an easy way to connect to SoundStructure systems.

Using the Address Book

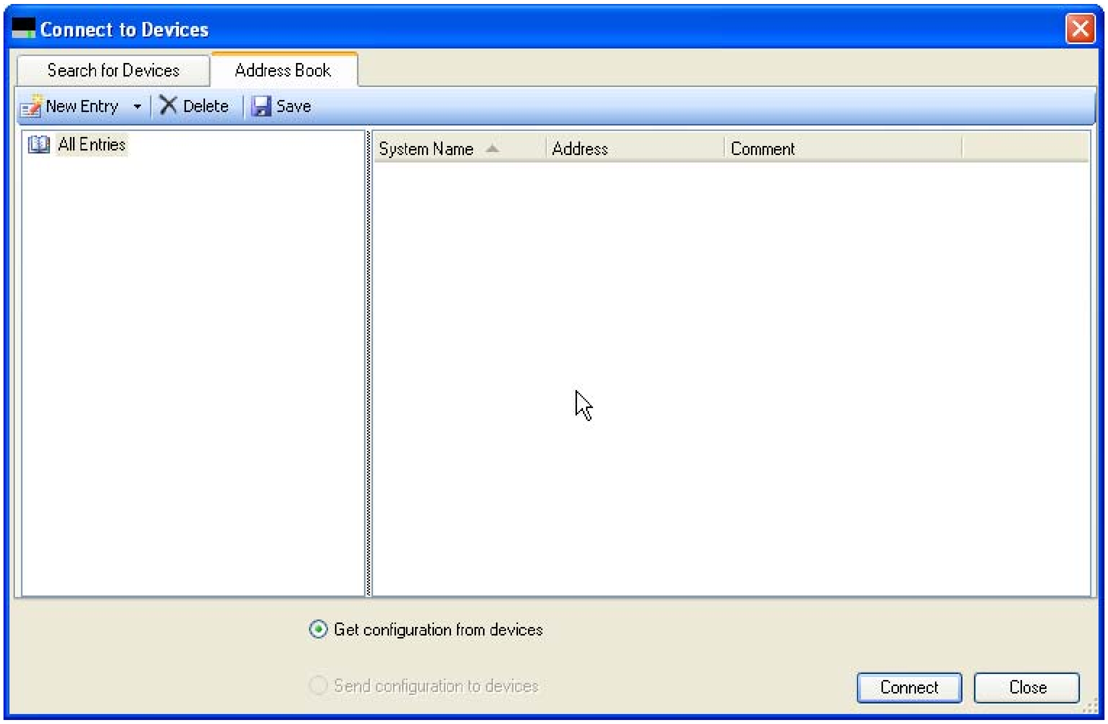
SoundStructure Studio version 1.2 and above includes an address book that makes it easy to connect over IP to SoundStructure systems across a network. The address book allows one to create and organize a collection of SoundStructure systems, storing the IP address, the system name, and a descriptive comment. A SoundStructure system may be a single device or multiple devices linked with OBAM.

While not a requirement, it is recommended that SoundStructure systems stored in the address book have static IP addresses. Systems with dynamic IP addresses may also be stored in the address book, however these systems may become unreachable if their IP address changes over time.

The address book is located under the Connect menu item as shown in the following figure.



The default address book is empty and will appear as shown in the following figure.


Connect to Devices
Search for Devices
Address Book
New Entry
Delete
Save
All Entries
System Name
Address
Comment
Get configuration from devices
Send configuration to devices
Connect
Close

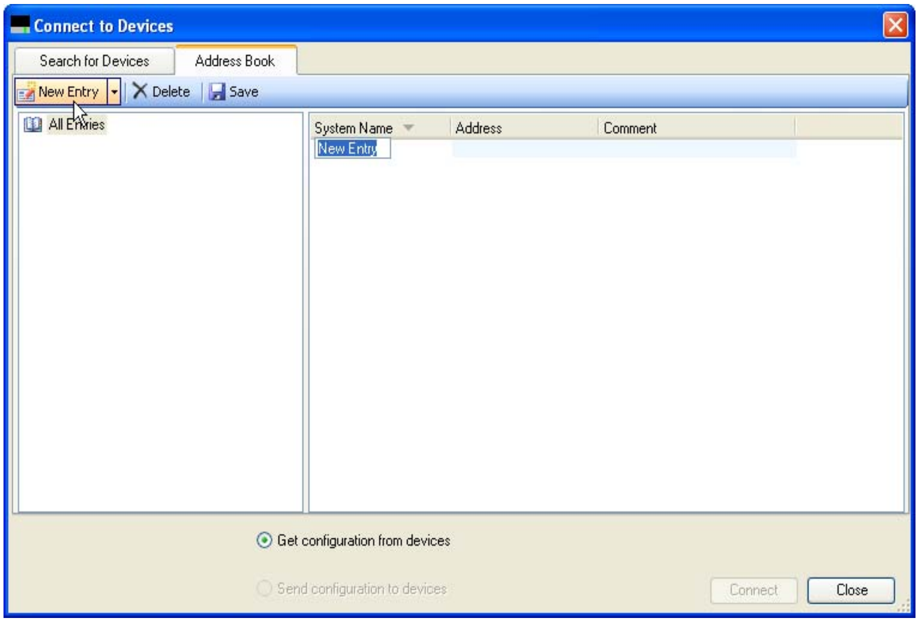
There is an item in the left pane called “All Entries” that shows all entries in the address book. By clicking on the desired column header, these entries can be sorted by System name, IP address, or Comment.

Address Book SoundStructure System Entries

The address book consists of one or more SoundStructure system entries. These entries may organized into folders as described in the following sections. The address book entries may either be added manually or added systems that SoundStructure Studio connects to may be easily added to the address book.

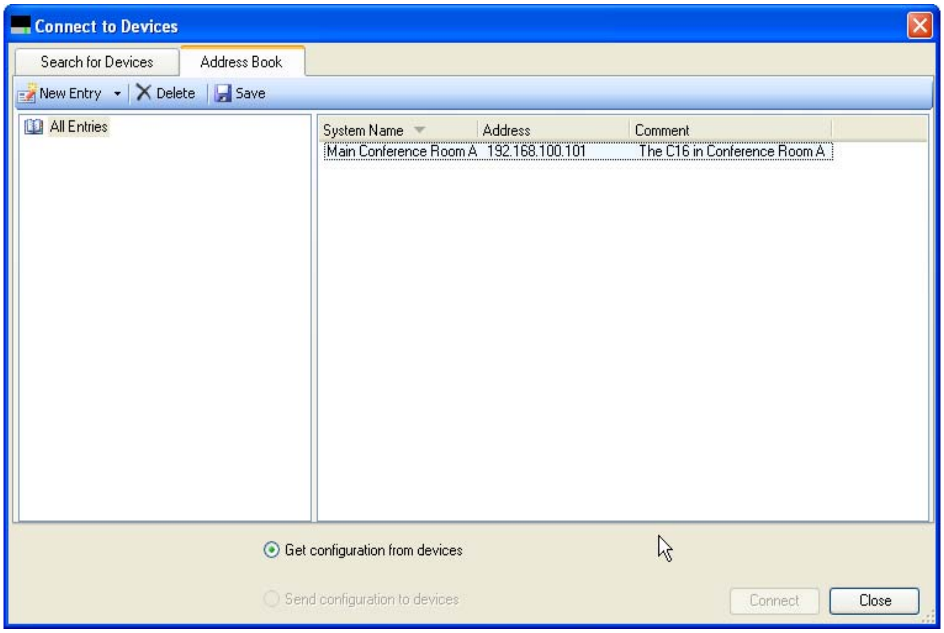
Creating Address Book Entries

To create an address book SoundStructure system entry, click the ‘New Entry’ menu item and edit the highlighted system name.


Connect to Devices
Search for Devices
Address Book
New Entry
Delete
Save
All Entries
System Name
Address
Comment
New Entry
Get configuration from devices
Send configuration to devices
Connect
Close

Once the system name has been added, you may either press Tab to go to the IP address field or single left click on the Address field associated with the system name and type in an IP address. A comment may be added by either hitting Tab or single left clicking on the comment field.

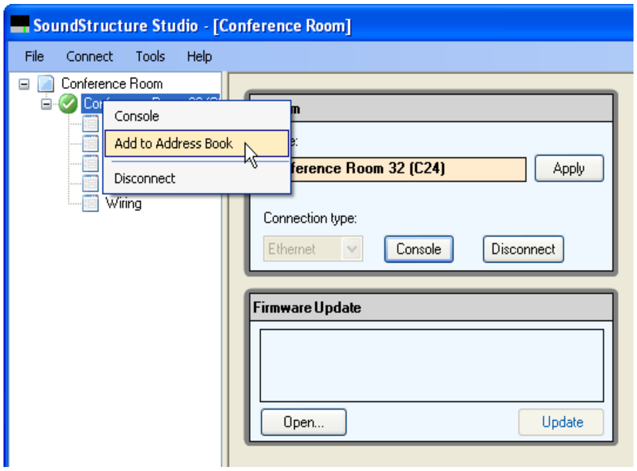
After an entry has been created, the address book will appears as in the following figure.


Connect to Devices
Search for Devices
Address Book
New Entry
Delete
Save
All Entries
Get configuration from devices
Send configuration to devices
Connect
Close

System Name	Address	Comment
Main Conference Room A	192.168.100.101	The C16 in Conference Room A

Adding Connected Systems to the Address Book

Once SoundStructure Studio is connected to a system, the system may be added to the address book by right clicking on the system name as shown in the following figure.

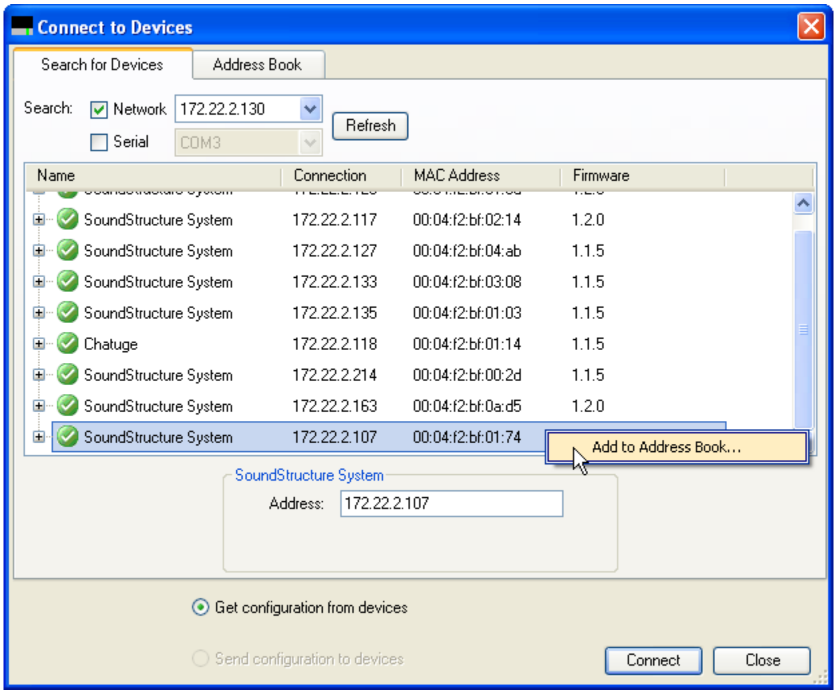

SoundStructure Studio - [Conference Room]
File
Connect
Tools
Help
Conference Room
Console
Add to Address Book
Disconnect
Wiring
Apply
Console
Disconnect
Firmware Update
Open...
Update

Connection type: Ethernet

ference Room 32 (C24)

The system and IP address will be stored into to the address book and raise the address book page to allow a comment to be added.

SoundStructure systems can also be added to the address book from the ‘Connect to Devices’ control by right clicking on the system name as shown in the following figure.


Connect to Devices
Search for Devices
Address Book
Refresh
Add to Address Book...
Connect
Close

Search: ☑ Network 172.22.2.130

☐ Serial COM3

Name	Connection	MAC Address	Firmware
SoundStructure System	172.22.2.117	00:04:f2:bf:02:14	1.2.0
SoundStructure System	172.22.2.127	00:04:f2:bf:04:ab	1.1.5
SoundStructure System	172.22.2.133	00:04:f2:bf:03:08	1.1.5
SoundStructure System	172.22.2.135	00:04:f2:bf:01:03	1.1.5
Chatuge	172.22.2.118	00:04:f2:bf:01:14	1.1.5
SoundStructure System	172.22.2.214	00:04:f2:bf:00:2d	1.1.5
SoundStructure System	172.22.2.163	00:04:f2:bf:0a:d5	1.2.0
SoundStructure System	172.22.2.107	00:04:f2:bf:01:74	

SoundStructure System

Address: 172.22.2.107

● Get configuration from devices

○ Send configuration to devices

Once systems are in the address book, it is possible to connect to those systems directly by highlighting the system name and clicking the *Connect* button.

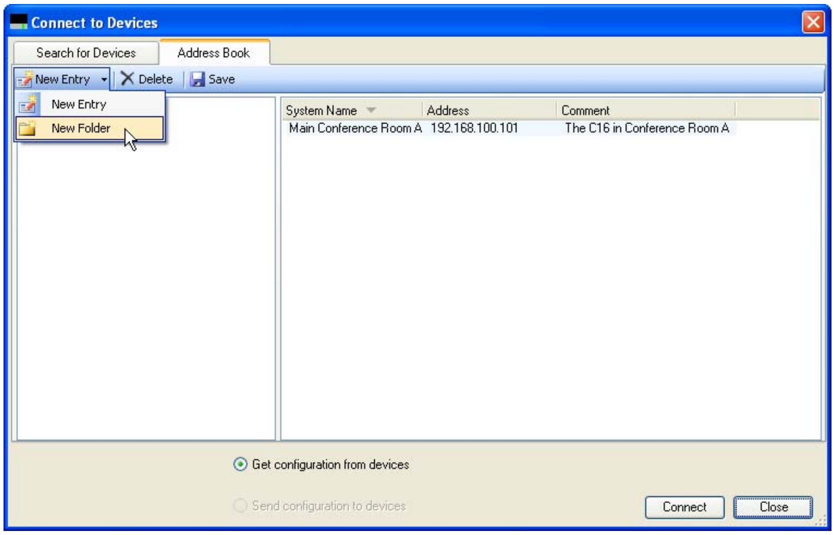
If any changes are made to the address book, an attempt to navigate away from the address book prompts the user to save changes to the address book. To preserve changes to the address book, answer Yes.

Address Book Folders

When an organization has multiple SoundStructure systems, it is useful to organize the systems with address book folders. Folders allow the user to organize the systems based on geography, or other criteria, so that only the desired systems are shown when the folder is selected. Folders may contain other folders.

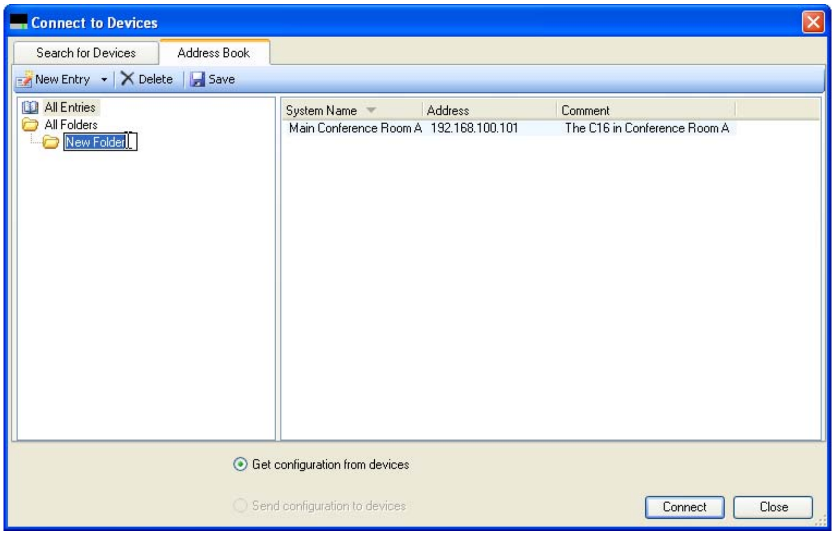
Within the address book, a folder contains a shortcut to the actual SoundStructure system entry and does not duplicate the data of the underlying entry. This means there is only one data entry for a SoundStructure system even though that system may be stored in multiple folders. Any change made to an entry's IP address, system name, or comment will be reflected in all folders where the system resides.

To create a folder, select 'New Folder' from he pull down 'New Entry' menu as shown in the following figure.


Connect to Devices
Search for Devices
Address Book
New Entry
Delete
Save
New Entry
New Folder
Get configuration from devices
Send configuration to devices
Connect
Close

System Name	Address	Comment
Main Conference Room A	192.168.100.101	The C16 in Conference Room A

Type in the name for the folder to overwrite the default name of New Folder as shown in the following figure.

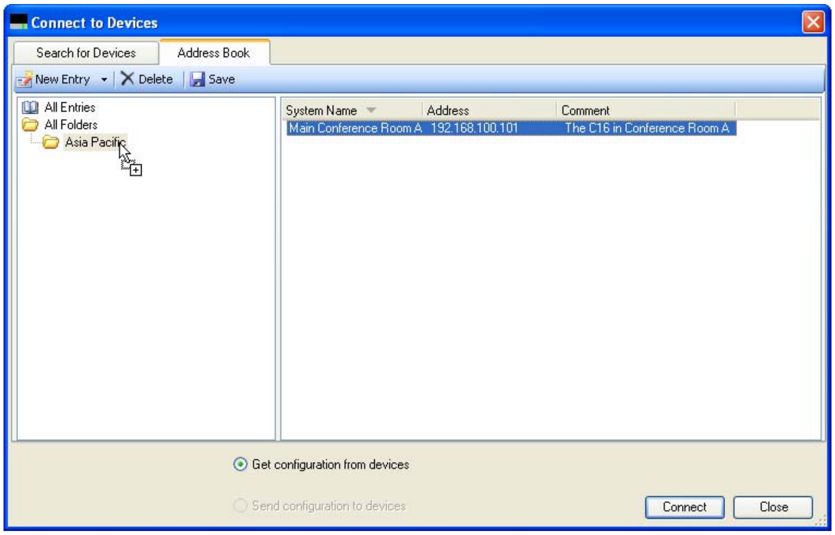

Connect to Devices
Search for Devices
Address Book
New Entry
Delete
Save
All Entries
All Folders
New Folder
Get configuration from devices
Send configuration to devices
Connect
Close

System Name	Address	Comment
Main Conference Room A	192.168.100.101	The C16 in Conference Room A

A new folder may also be created by right clicking on the top level folder and selecting a new folder. Select the 'All Entries' item to see all the SoundStructure systems in the address book.

Adding systems to a folder

A SoundStructure system entry may be *copied* to a folder by clicking and dragging the system from the All Entries area to the desired folder as shown in the following figure. The “+” symbol near the cursor indicates the system is being copied to the folder.



Connect to Devices
Search for Devices
Address Book
New Entry
Delete
Save
All Entries
All Folders
Asia Pacific
Connect
Close

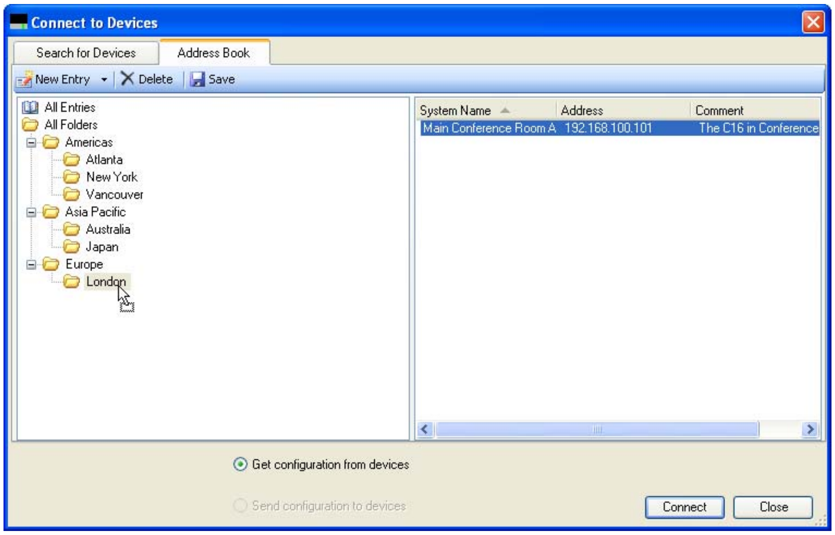
System Name	Address	Comment
Main Conference Room A	192.168.100.101	The C16 in Conference Room A

● Get configuration from devices

○ Send configuration to devices

Moving systems from one folder to another folder

A SoundStructure system may be *moved* from one folder to another folder by clicking and dragging the system from the source folder to the target folder as shown in the following figure. In this example the system from a source folder was clicked and dragged to move it into the London folder. There is no “+” symbol by the cursor when systems are moved.



Connect to Devices
Search for Devices
Address Book
New Entry
Delete
Save
All Entries
All Folders
Americas
Atlanta
New York
Vancouver
Asia Pacific
Australia
Japan
Europe
London
Connect
Close

System Name	Address	Comment
Main Conference Room A	192.168.100.101	The C16 in Conference

● Get configuration from devices

○ Send configuration to devices

Copying systems from one folder to another folder

A SoundStructure system may be *copied* from one folder to another by holding the Control key and clicking and dragging the system from the source folder to the target folder. The cursor will show a “+” symbol to confirm that the system is being copied from one folder to another folder.

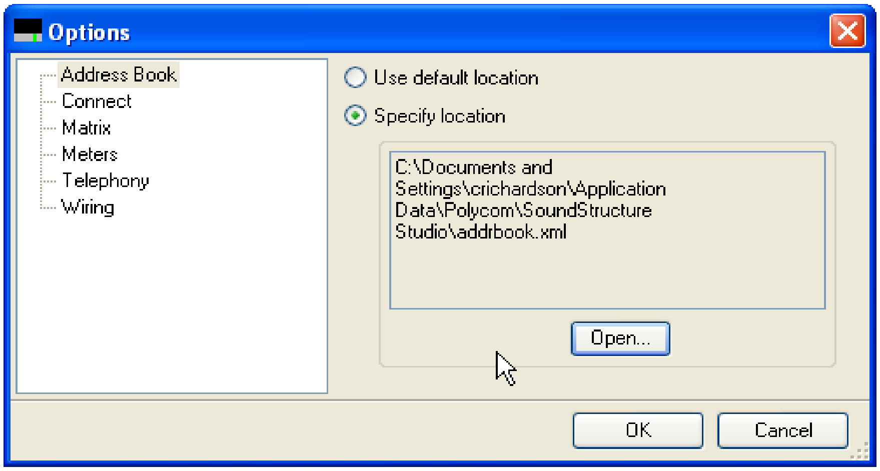
Removing Entries from the Address Book

Entries may be deleted from a folder by selecting the entry and pressing the delete key on the keyboard or hitting the Delete button. Deleting an entry from a folder removes the entry from the folder but does not delete the underlying entry.

To remove an entry from the address book entirely, click on the All Entries icon, select the target system, and delete the system. This will permanently delete the entry from all folders that this system belonged to and will also remove the system from the address book.

Changing the Location of the Address Book

By default the address book is stored on the local computer in a directory that may be found under the **Tools** > **Options** > **Address Book** . The default location on the local drive may be changed by selecting the ‘Specify Location’ option and opening the target address book as shown in the following figure.



○ Use default location

● Specify location

C:\Documents and Settings\crichardson\Application Data\Polycom\SoundStructure Studio\addrbook.xml

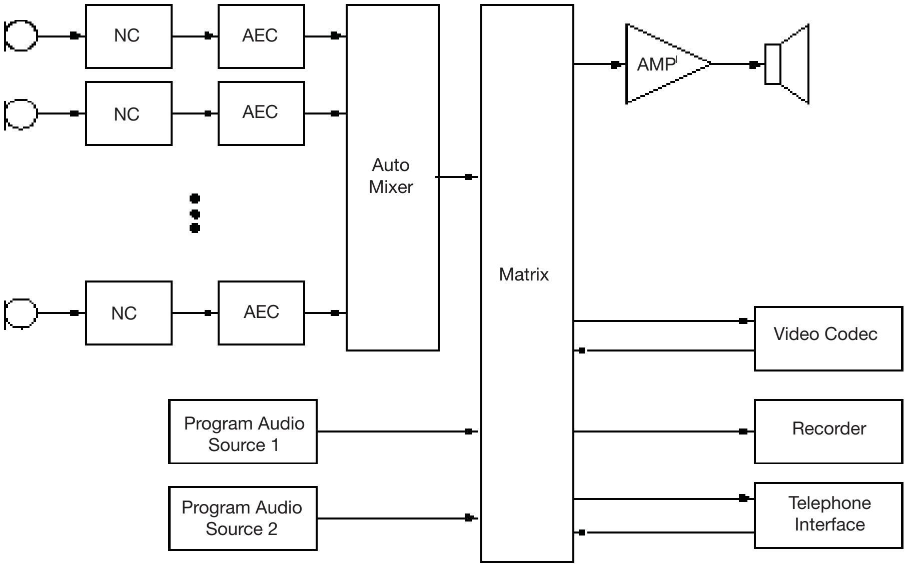
It is possible to create an address book on one computer and then use that address book as a master address book for an enterprise by storing the address book on a shared drive and have all users point to that file for their address book.

Appendix C: Designing Audio Conferencing Systems

Reprinted from the BICSI AV Design Reference Manual, the following audio conferencing design material is authored by Craig H. Richardson, copyright 2006.

The goal of audio conferencing is to enable two geographically separated groups of people, referred to as the local and remote talkers respectively, to communicate as if they are in the same room together with fast interation and allowing both parties to speak and be heard at the same time.

The following figure shows a typical solution for audio conferencing (or audio for video conferencing) that consists of local microphones and loudspeakers, an acoustic echo canceller and noise canceller, automatic microphone mixers, matrix mixers, telephony interfaces, video codecs and possibly some program audio (such as CD's, DVD's, or video tapes). The rest of this chapter will explore the different aspects of the following figure in more detail. This type of configuration would be installed in each of the rooms that are conferenced together.



The challenge in audio conferencing is that the loudspeaker audio is not only heard by the local participants, but it is also heard by the local microphones and, in the absence of an acoustic echo canceller, will be sent back to the remote participants causing the remote participants to hear a delayed echo of their voice.

Many of the challenges in audio conferencing arise from microphone selection and placement, loudspeaker placement, and balancing the architectural and aesthetics of the room with the requirements for having the best audio quality. Compromises in audio quality in favor of aesthetics will reduce the quality and intelligibility of meetings.

In small conferencing room spaces, a tabletop conferencing phone is often an adequate solution for audio conferencing and provides an all-in-one package that is easy to deploy and provides good sound quality in the many room environments.

Large Room Environments

The typical room environment introduces ambient noise (from HVAC, outside noise, projectors, computers, etc.), in-room reflections of the audio (multipath audio), and constraints on microphone and loudspeaker positioning.

Conference rooms should be designed to a NC 30 standard if possible. An NC 30 rating corresponds to approximately 40 dBA SPL of background noise, leaving about 30 dB of SNR for normal talkers (70 - 77 dBA SPL at 1 meter). A lower NC rating will further improve the quality of the audio conferencing system by improving intelligibility and reducing listener fatigue but will also increase the overall cost of the room build-out. For an audio system to have good intelligibility, it is important for the signals that are heard by the local talker to be at least 25 dB above the background noise level.

The surfaces in the room including walls, ceiling, and furniture will affect the quality of the conferencing system. Hard surfaces will increase the amount of reflections in the room causing multiple versions of a local talker's audio signal to be heard by the microphones and for multiple versions of the remote talker's signal to be heard in the local room. Multiple versions of the signals that are time delayed with respect to each other will cause noticeable comb filtering effects that will filter out particular frequencies (dependent on the time separation of the multiple versions of the signals), degrading the quality of the signal. Once the frequencies are filtered out by the comb filtering, they can not be restored by equalization.

Often the conference room location is selected due to the outside view from the room or because it is a convenient location. However the location of a potential conference room should be evaluated to make sure it is not directly underneath a building's HVAC units, nor near other environmental noise sources such as shipping rooms, loading docks, copy rooms, network operations centers, and other such rooms to ensure that the outside noise sources are minimized in the conference room.

Microphone Selection And Placement

The type of microphones used and their location will have the largest impact on the audio conferencing quality. Microphones translate the acoustic signals from the local talkers into electrical signals that can be processed and sent to the remote participants.

Microphone Fundamentals

Most microphones used in conferencing systems are electret microphones, a version of condenser style microphones where an acoustic signal on a thin film dynamically varies the capacitance of an electrical circuit which in turn creates an electrical voltage that represents the microphone signal. Condenser microphones require a bias voltage, called phantom power, to operate properly. Electret microphones are a variant of condenser microphone that replaces the thin film with a dielectric material that is permanently charged and suspended above a metal plate. While electret microphones don't require a bias voltage to operate due to being permanently charged, they do typically contain an integrated preamplifier that is powered using the phantom power from the device the microphone is connected to. Due to the design of electret microphones, these microphones come in a large variety of sizes and shapes and can provide excellent audio quality.

Phantom Power

Electret (and condenser) microphones require a power supply, called phantom power, to power the electronics of the microphone. This power supply may come from a battery or from the electronics that the microphone is connected to. Microphones typically operate with phantom power voltages ranging from 9 to 54 V DC (with 48 V specified in the standard IEC 61938) although there are some microphones that only operate with a more limited range and will not operate with 48 V. Electret microphones typically require approximately 2 mA of current of phantom power although they can require as much as 10 mA.

The phantom power is supplied across the positive and negative balanced audio signals with respect to the ground/shield of the microphone.

Directional vs Omni-directional Microphones

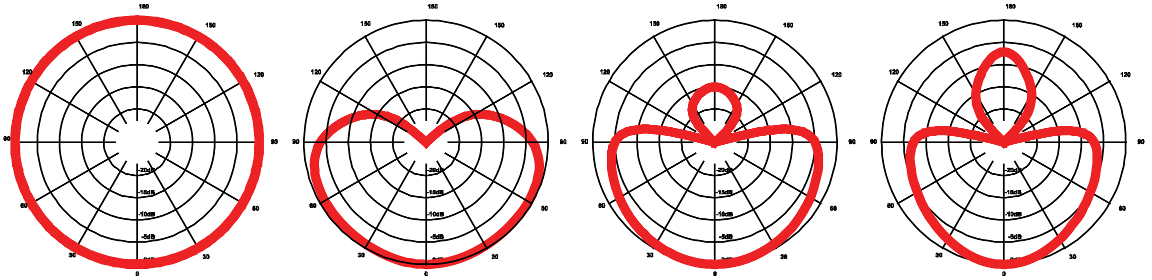
Electret microphones are either omni-directional and directional in their pick-up pattern depending on how the electret microphone element is physically mounted within the microphone enclosure.

Omni-directional microphones will pick up sounds from all directions around a microphone (a 360 degree pick up pattern) while directional microphones have been designed to pick up signals better in the pick up zone of the microphone and to reject signals outside of the pick up zone.

Directional microphones are most often used in conferencing systems due to the rejection of the background noise, reduction of the reverberation, and the rejection of the audio from the loudspeakers. Directional microphones also increase the gain-before-feedback in sound reinforcement applications due to the increased rejection of the loudspeaker signal when it is directed towards the rear of the directional microphone.

As seen in the following figure, directional microphones typically have three degrees of directionality - cardioid, super cardioid, and hypercardioid - each with increasing directionality (that is, decreasing areas of pick-up). A microphone with a cardioid pick up pattern will pick up signals within about 130 degrees of the front of the microphone, a super cardioid about 115 degrees, and a hypercardioid about 105 degrees. From the following figure it is clear that the supercardioid and hypercardioid microphones have a lobe in the rear of the microphone that will pick up background noise more than the cardioid microphone.

Shotgun microphones are even more directional that hypercardioid microphones and attain this directionality by having the microphone element in a tube with slots cut along the side. The signals from a narrow arrival angle will reinforce constructively while those from off angle that travel farther will cancel each other out, providing an extremely narrow pick up pattern. Shotgun microphones are typically not used in conferencing due to the extremely narrow pick-up pattern - should the local talker move out of the narrow pick up pattern, their voice will be significantly attenuated. To the remote listeners, it will sound like the local talker has faded out.



Noise Immunity

Cellular telephone systems often have transmit and receive frequencies in the 900 to 1800 MHz range. During normal operation of these cellular phones there is regular communication with the closest cellular towers for status as well as signaling incoming data and telephone calls. For some GSM style phones this signaling occurs at the rate of one 576 µs pulse every 4.6 ms which represents a 217 Hz signal. Often in conferencing applications, these cellular telephones are placed on the conference table in close proximity to the local microphones. Because these microphones often are not designed to be immune to frequencies in the 900 MHz to 1800 MHz range, that the transmit and receive signal can be easily coupled into the microphone where it is rectified by microphone's electronics and combined with the local microphone audio signal. When this happens, a noticeable beeping or chipping sound that sounds modulated such as if it were Morse code, will be heard at the remote locations.

If this problem is present, the solutions are to move cellular telephones away from the microphones, turn off cellular telephones, or to use microphones that have improved noise immunity to these high frequencies. Many manufacturers are now producing microphones with improved noise immunity.

Boundary Microphones

Boundary microphones use the surface the microphone is installed on and the proximity of the microphone element to the boundary surface to minimize the amount of phase cancellation that occurs when audio strikes the boundary. The resulting microphone configuration has a higher sensitivity. The pick-up pattern of the microphone will become half-spherical as the sounds below the boundary are not picked up. For instance if an omni-directional microphone is placed on a boundary, the pattern becomes semi-spherical. A cardioid microphone placed on a boundary will become a semi-cardioid pattern with the sound below the boundary not picked up by the microphone.

Critical Distance

For every audio source in a room there is a distance from that source, called the critical distance, where the reverberant sound field and the direct sound field from the source are equal in intensity. If a microphone is placed farther than the critical distance away from the source, typically a local talker, the resulting speech quality will be considered very poor - characterized by a bottom-of-the-barrel or muffled sound. The critical distance is a function of both the physical distance from the local talker to the microphone, the directionality of the source, and the liveliness of the acoustics in the room. More reverberant rooms will have a shorter critical distance which underscores the requirement to place microphones as close to the talkers as possible. Increasing the gain on a microphone will not help reduce the critical distance as the reverberation and noise will be amplified along with the local talker's voice when the gain is increased.

The critical distance can be measured with an SPL meter and noise source. When the measured sound level doesn't drop by 4 to 6 dB for each doubling of the distance, the critical distance of the microphone from the noise source has been reached.

As a rule of thumb, for omni-directional microphones, the microphone should be no farther than 30% of the critical distance away from the talker. A directional microphone should be placed no farther than 50% of the critical distance.

If due to architectural constraints, or room usage requirements, the microphones must be placed farther than 50% of the critical distance (for instance with ceiling microphone installations), the users must either accept the resulting speech quality or increase the effective critical distance by moving microphones closer to the talkers, moving noise sources away from the microphones, lowering the level of the noise, and improving the acoustics in the room to reduce the amount of reverberation. Improving the acoustics in the room can be done by increasing the absorption of surfaces in the room - acoustic paneling, reduced HVAC airflow speed/noise, carpeted floors, curtains, and other absorptive surfaces wherever possible.

Microphones For Conferencing

While omni-directional microphones seem like a natural choice for conferencing applications as fewer microphones would be required for a given number of participants, their 360 degree pick up will pick up extra noise, room reverberation and the remote audio from the loudspeakers. The result will be that the signal picked up by the microphone in a conferencing environment will sound muddier and noisier than the signal from a directional microphone. It is for this reason that most installations use cardioid style microphones.

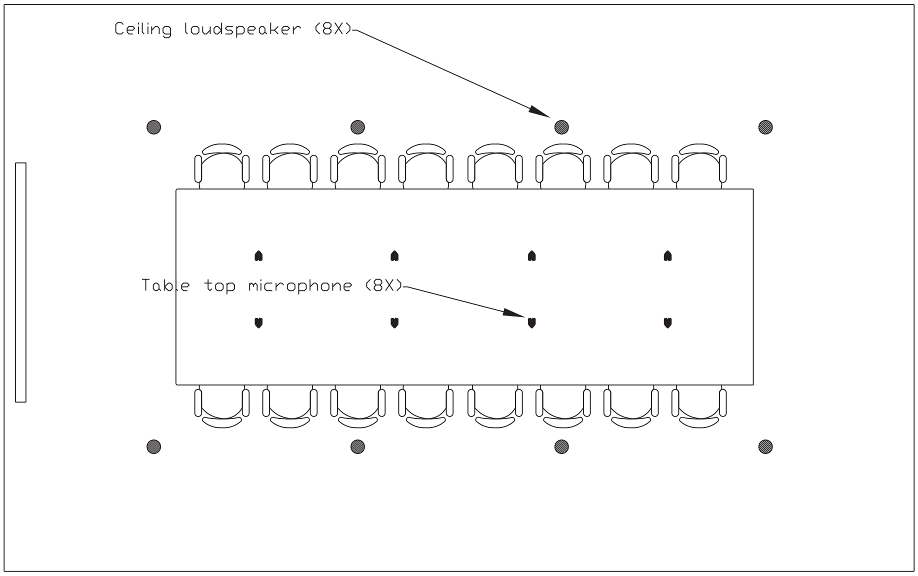
The style of microphone is determined by how the participants will use the conference room. It is common in rooms where there is a great deal of collaboration or sharing of drawings, documents, etc. to get the microphones off the table and have them in the ceiling. This allows for much sharing of paperwork without the concerns or complaints of the sounds of paper as it slides across the microphone elements.

Gooseneck Microphones

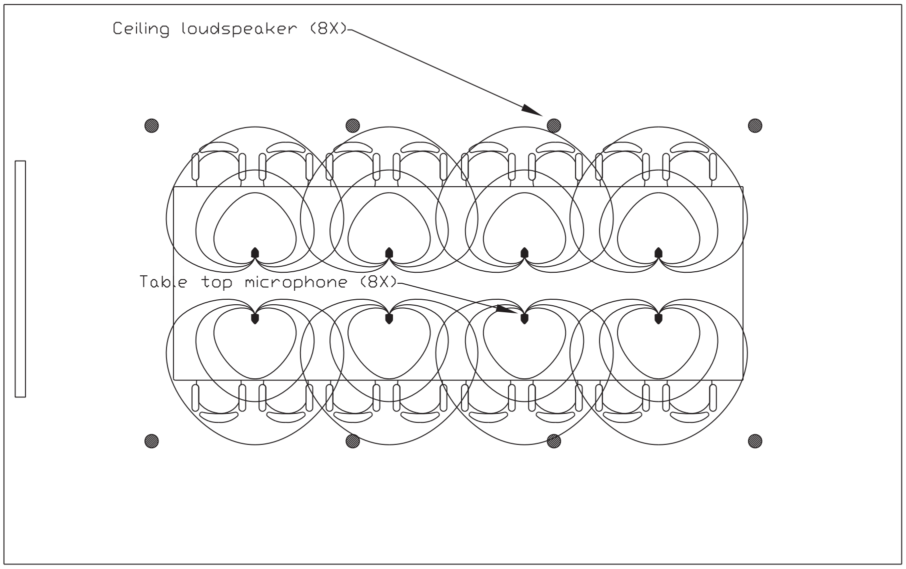
In many board rooms, gooseneck microphones are used and distributed around the table with one microphone per participant. Each of these microphones will have a neck that takes the microphone off the table and brings it closer to the local talker's mouth. In addition these microphones will have a directional pattern to further improve the rejection of the ambient noise. One consideration to remember is that some gooseneck microphones have a tendency to straighten-out over time, so it is recommended that the microphone positioning be checked periodically to ensure that the gooseneck microphones are not pointed directly at loudspeakers.

Tabletop Microphones

In many conferencing applications boundary tabletop microphones are used and mounted at locations around the table as shown in the following figure where one microphone is used for each two participants.



All participants that are seated at the table are well within the microphones pick-up ranges as shown in the next figure.



Wireless Microphones

Wireless microphones are also commonly used in conferencing applications, particularly for presenters. The advantage of wireless microphones is that they may be used anywhere within the room (depending on whether in-room sound reinforcement is used with that microphone) and have the advantage that they are usually closer to the local talker's mouth than a tabletop microphone. It is better to use a lavalier style of wireless microphone than a hand-held wireless microphone to ensure that the microphone stays a constant distance from the talker's mouth which will minimize the sound level variation to the remote participants in a conferencing application. Wireless handheld microphones typically end up being placed on tables, used as pointing devices, and end up in other situations that reduce the quality of the resulting audio signal. One consideration with lavalier microphones is to ensure the users of the microphones understand where to attach the microphone to minimize the occurrence of fabric being dragged across the microphone element.

There are different methods of transmission that wireless microphones use, ranging from analog transmission over UHF or VHF frequencies to digital encrypted transmissions over a 2 GHz frequency band. Considerations for choosing a wireless microphone include the signal bandwidth that is transmitted (narrow or wide band), the latency introduced by the wireless microphone, the battery type and life (rechargeable or not), reliable transmission distances, cost, freedom from interference, and whether it uses encryption to secure the wireless channel.

Ceiling Microphones - Quality vs Convenience

In all applications, getting the microphone closer to the local talker results in a higher quality signal due to improved signal-to-noise ratio (the local talker is louder compared to the background noise) and a reduction in the amount of the multipath signal compared to the original signal.

In most conferencing applications there is a trade-off with the usage of a room, the location and configuration of the furniture, and the maximum achievable audio quality. It is common for rooms with movable furniture to not have permanently installed microphones on the tables. To accommodate this, microphones are often moved away from the participants and placed in the ceiling. A consequence of this is that the resulting signal picked up by the microphone includes significant amount of additional noise sources that are closer to the microphone than the talker such as HVAC. In addition, the local talker's mouth will be farther from the microphone, reducing the signal level that is picked up by the microphone. Any additional gain added to pick up the local talker's signal will also amplify the background noise, exacerbating the problem.

To set proper expectations, demonstrate the audio quality of a ceiling microphone in the target room as early in the design process as possible. This can be done simply by temporarily hanging a microphone from a ceiling with tape or some other temporary adhesive, and running the microphone cable to a location acoustically isolated from where the microphone is temporarily installed. With the use of a microphone amplifier and powered loudspeaker, it is a simple matter to have the customer listen to the audio quality and agree that it is acceptable before additional work is performed. This will save costly re-installation work. More often than not, once the conference room users hear the resulting sound quality of ceiling microphones, they will allow tabletop boundary microphones to be used in their rooms, or at the very least agree that the microphones must be placed closer to the local participants.

When using ceiling microphones, it is important to install the microphones away from noise sources in the ceiling including HVAC, light fixtures, projectors, loudspeakers, and any other noise source. If ceiling microphones are the only option, it is better to hang the microphone from the ceiling as close to the talkers rather than to have them flush-mounted if possible. While aesthetically less appealing than a flush-mounted ceiling microphone, the hanging microphone will move the microphone away from any ceiling noise sources including the loudspeakers that will play the remote audio into the local room and get the microphones closer to the talkers. If the ceiling microphone is close to the ceiling (less than a foot from the ceiling), but not flush mounted, the same boundary affect that improves the sensitivity of flush-mounted microphones will cause multiple delayed signal paths to be picked up by the microphone, unnecessarily degrading the overall audio quality of the system. Ceiling microphones (hanging or flush mounted) should have no more than a half-spherical pick up pattern. In typical conference rooms with 9 foot high ceilings, it is common to hang the ceiling microphones one or more feet from the ceiling over the conference table. Again, the closer to the talker's mouths the better the system can sound.

When planning for how many ceiling microphones are required, keep in mind that a ceiling microphone will cover approximately 100 sq. feet of room space. This is dependent on the critical distance of the microphones. For instance, rooms that have very high ceilings (defined as > 10 feet) should only use ceiling microphones if it is possible to hang them several feet from the ceiling.

Automatic Microphone Mixers

As described elsewhere in this guide, the role of the automatic microphone mixer is to limit the number of microphones that are open (or contributing audio) at any given time to only the microphones associated with active talkers. By reducing the number of microphones that are active, the local speech that will be sent to the remote site will be less reverberant and less noisy.

There is a significant reduction in audio quality if all the microphones are active all the time versus being processed by the automatic microphone mixer. This difference is particularly obvious when ceiling microphones are used due to the high noise and reverberation levels associated with microphones that are farther away from the local talkers and approaching greater than 50% of the microphone's critical distance.

Noise Cancellation

The ambient noise in the room caused by HVAC, projectors, computers, and even noise external to the room that is picked up by the microphones will reduce the signal to noise ratio at the microphones. This noise will then be transmitted to the remote site along with the local talker's audio signal and the reduced signal-to-noise ratio will contribute to lowered intelligibility of the remote audio and increased listener fatigue for the remote talkers.

It is best to eliminate or at least reduce the ambient noise through architectural means such as changing HVAC ductwork, moving microphones away from noise sources, and removing or dampening noise sources. If these approaches are not adequate or possible, an additional option is to process the microphone signal with advanced signal processing techniques that reduce the level of background noise while maintaining the quality of the local talker's voice.

Techniques for reducing the background noise picked up by the microphone range from simple noise gates to advanced digital adaptive filters. Noise gate techniques will reduce the noise when the local talkers are not talking by suppressing the signal that is below a given threshold, but the noise will still be present when the local talkers begin to speak again. The gating of the background noise will sound unnatural at the remote site as the local talker speaks and then stops speaking.

More sophisticated techniques such as adaptive filter techniques are used quite successfully in audio conferencing applications. While not all adaptive noise reduction techniques (commonly referred to as noise cancellation) have the same performance, the objectives are the same - to first identify the characteristics of the noise (broadband such as HVAC noise, or narrowband such as a whine from a mechanical source) and then remove that noise signal from the microphone audio signal without any additional information about the noise. These techniques work best with noise that has stationary statistics -for instance, the noise signals may be random, but the style of randomness is fixed such as the noise from a fan source. As these techniques typically take several seconds to identify the characteristics of noise, these techniques do not work well with impulsive noises such as clicks from pen tapping or paper rustling on a microphone. These systems are typically designed to work with speech signals and are not usually suitable for use with music.

As not all implementations are the same, there can be a large variation in the amount of residual noise or spectral artifacts that are introduced into the processed signal. These artifacts can sound like chirps or worse and may be perceived to be worse in quality than the original noise. These artifacts may be minimized by lowering the amount of noise cancellation provided - typically it can be adjusted from 0 to 15 dB or more. With current techniques 5 -10 dB of noise cancellation can be achieved without significant distortion of the underlying local talker's signal (depending on the manufacturer).

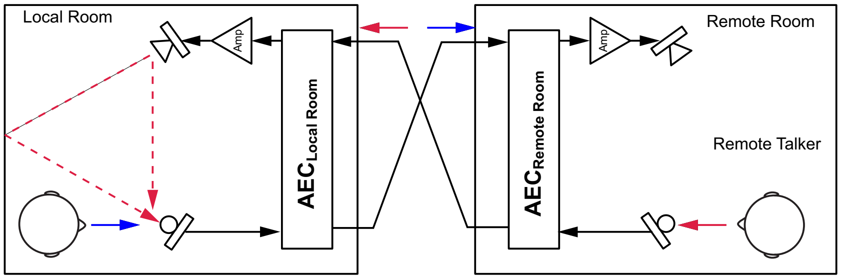
Ceiling microphones benefit the most from noise cancellation techniques as these microphones are closest to the ceiling noise sources of HVAC and projectors. The noise cancellation can make an otherwise useless room usable. However, if ceiling microphones are swaying due to the air flow from nearby HVAC ducts, noise cancellation may not be able to completely remove that noise.

Acoustic Echo Cancellation

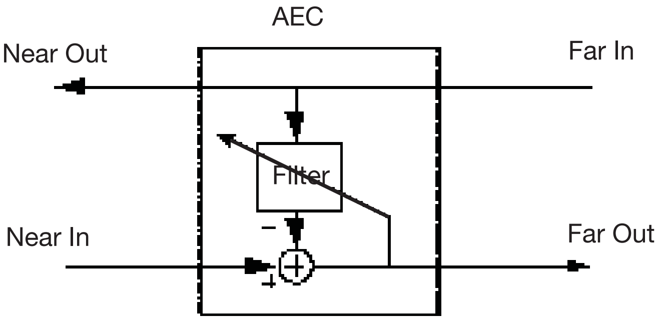
In audio conferencing applications, acoustic echoes occur because an open-air acoustic path exists between the local loudspeaker and the local microphone. As shown in the following figure, speech originating in either the local or remote room is transmitted over a communications network to the other room where it is amplified and reproduced by the local loudspeaker. The output of the loudspeaker (or multiple loudspeakers) will fill the local room and, from many paths of reflections, reach the microphones in the local room. In the absence of an acoustic echo canceller, this acoustically-echoed version of the remote talker's audio is transmitted back over the network to the originating room and is reproduced by the loudspeaker where it is perceived as an acoustic echo.

Acoustic echoes degrade the quality of speech communications because echoes of one's speech are subjectively annoying to the person speaking. In fact, if the elapsed time between when a word is spoken and when its echo is heard is more than 300 msec, the echo will actually cause most talkers to stutter. In addition, echoes can overload communication circuits, resulting in a feedback condition called howling. If, by the combination of loudspeaker volume and microphone sensitivity and positioning, the echoes are louder than the originating speech, the teleconferencing equipment, or the network itself, can overload.

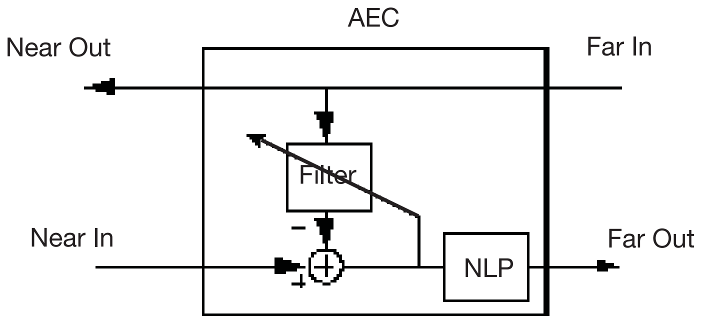
An acoustic echo canceller in the local room will remove the echo of the remote talkers' voice so it will not be sent back to the remote talkers. Just as an acoustic echo canceller is used in the local room to prevent echoes to be sent to the remote participants, the remote site would have a similar audio conferencing solution to prevent the local talker's audio from echoing back from the remote room as shown in the following figure.



The sophisticated part of an acoustic echo canceller consist of an adaptive filter that is nearly continuously adapting to adjust to changes in the echoes in the room caused by changes in the local room such as participants moving around. The following figure illustrates the block diagram of an AEC system and shows the adaptive filter at its center. It is common to refer to the signals associated with an echo canceller as follows: the Near In signal consists of the local microphone audio (both local talker and the echo of the remote talker), the Near Out is the signal that will be played into the local loudspeaker so that the local participants can hear the remote participants. The Far In is the signal sent from the remote side and Far Out is the local talker audio with the acoustic echo-removed that is sent to the remote participants.



The performance of the AEC is often improved with the addition of non-linear processing (sometimes referred to as center-clipping) applied after the adaptive filter on the Far Out signal. The non-linear processing will remove low-level artifacts that result from imperfections of the adaptive filter, that is, when it is not completely adapted. Acoustic echo cancellers typically will have a user adjustable control for the amount of non-linear processing, allowing the processing to be more aggressive (larger amount of suppression) or less aggressive (less suppression). More aggressive suppression will reduce the occurrence of residual acoustic echoes to the remote site, but possibly at the expense of introducing some clipping of the first syllable of local talkers audio during transitions from the remote people talking to the local people talking. Aggressive suppression may also vary the level of the transmit audio signal based on whether both the local and remote talkers are speaking at the same time. With less aggressive suppression it is possible to make the system appear more full-duplex at the expense of potentially having some residual echo transmitted to the remote site if the acoustic echo canceller is not fully converged.

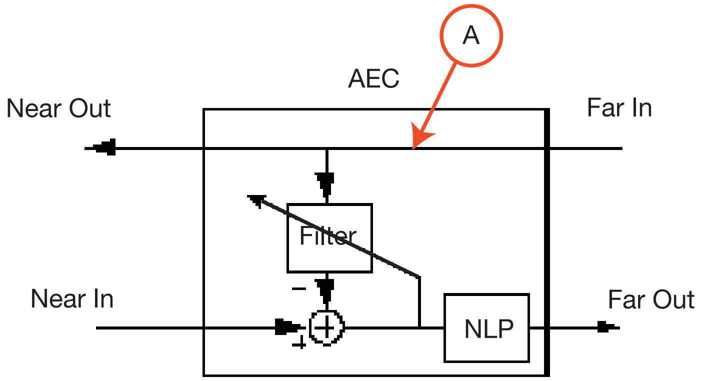


AEC Reference

The AEC reference provides the AEC with the information of what signals it should cancel from the room (see the following figure). The echo canceller reference is usually a combination of audio from the remote sites including telephone and video conferencing audio and also any program audio sources. Microphone signals are typically not included in the acoustic echo canceller reference signal.

In order to cancel echoes, the original source signal (the remote audio) must be part of the echo canceller reference. Generally the reference is the original signal that is played out of the loudspeaker system (as shown in the following figure) before it generates echo in the local room.

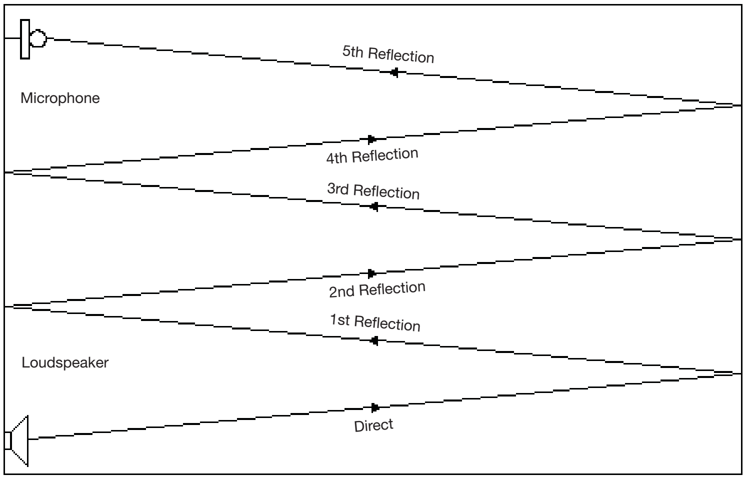
If the reference signal is significantly different from signal played out of the loudspeaker (due to dynamics processing, for instance), the echo canceller may not be able to cancel the echo signal because it is not recognized as being the same as the reference signal, causing a persistent residual echo to be sent to the remote site as the local echo canceller treats this signal as a local talker. Do not apply dynamics processing or other non-linear signal processing on the loudspeaker signal as that will distort the echo signal substantially from the signal the echo canceller is expecting to see for the echo canceller reference. If dynamics processing is required, process the signal before it is used as the echo canceller reference.



Tail Time

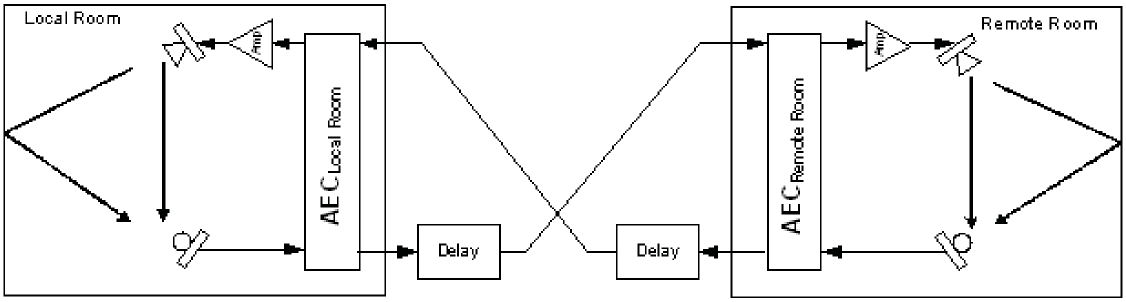
The tail time of an echo canceller is the amount of time the echo canceller can still recognize a signal as an echo from the remote talkers' speech and not interpret that signal as speech from a local talker.

All echo cancellers have some upper limit, called the tail time, after which they do not recognize the echo signal as a version of the reference signal and consequently can not remove the echo from the room. The tail time is measured in milliseconds (although can be interpreted in terms of distance) and should be greater than 100 msec for medium sized rooms and greater than 200 msec for larger rooms. As shown in the following figure, if the room is lively, the length of an echo path may be longer than expected making the room appear acoustically larger than it is physically.



Transmission Delay

In all communication networks, there is a combination of processing latency and network latency that delays when the local signal is heard in the remote room and conversely when the remote signal is heard in the local room. There have been numerous studies that have shown that when delay is added to an echo, the perceived communication impairment caused by the echo is increased. Processing latency is due to the selection of algorithms, the speed of the underlying processors, the implementation of the algorithms, and how the audio is collected from the analog-to-digital converter and also sent to the digital-to-audio converter in a digital system. In some cases, such as in video conferencing systems, delay may be intentionally added to the audio signal to compensate for the delays inherent with video signal in order to maintain audio and video lip synchronization. Network transmission delay is limited by the speed of electrons (or the speed of light with satellite and microwave transmission) in a network and any other processing or data handling that may occur in the data as it is transmitted through the network. It is not uncommon for networks to require a hundred milliseconds or more to transfer audio from one site to the other.



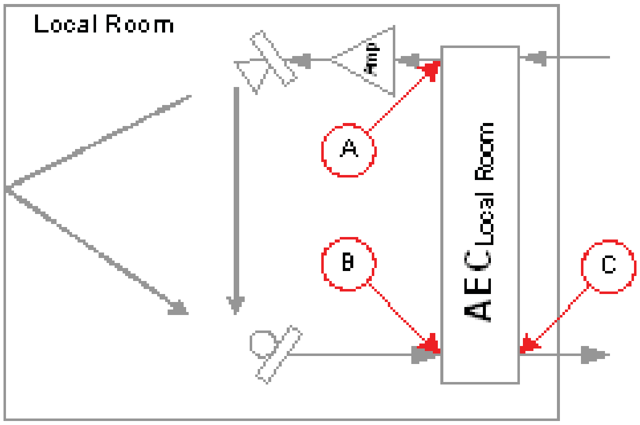
While the delay in the network will increase the perception of echoes, it does not affect the performance of the acoustic echo canceller in either the local or remote rooms. The acoustic echo canceller is concerned only with the audio as it enters the room and reflected in the room and picked up again by the local microphones. It doesn't matter to the echo canceller whether the signal from the remote talkers takes 1 millisecond or 10 hours to be received - the local echo canceller is only concerned with the audio once it reaches the acoustic echo canceller and picked up by the local microphones. It is only the perception of residual echoes that is influenced by the transmission delay. For instance an echo that is heard with 5 msec of delay will not be objectionable at all, while the same echo heard with 500 msec delay will render interactive communications impossible.

Echo Return Loss

An echo canceller's performance is measured by how well it can reduce the echo signal that is present at the microphone. The natural reduction of the echo signal due to the physical separation of the loudspeakers from the microphones is commonly referred to as echo return loss (ERL) and is measured in dB. As shown in the following figure ERL is the ratio of 10 Log(A/B) where A is the signal that is sent to the loudspeaker amplifier, and B is the signal picked up by the microphone. It is common for echo cancellers to have a minimum required ERL for proper operation - exceeding the required ERL will slow or prevent the acoustic echo canceller from properly converging. Typical values for ERL are 0 - 10 dB with 0 dB characterizing a more robust echo canceller than one that requires 10 dB of ERL for proper operation.

While the ERL reduces the amount of echo present at the microphones, there will still be a significant amount of echo that will be sent to the remote site if there is no further processing on the signal. The enhancement of the echo return loss due to the presence of an acoustic echo canceller is referred to as the echo return loss enhancement or ERLE. In the following figure, the ERLE would be the ratio of 10 Log (B/C) which, due to the acoustic echo canceller, should be a larger number than the ERL. Typical values for ERLE are 15 - 25 dB.

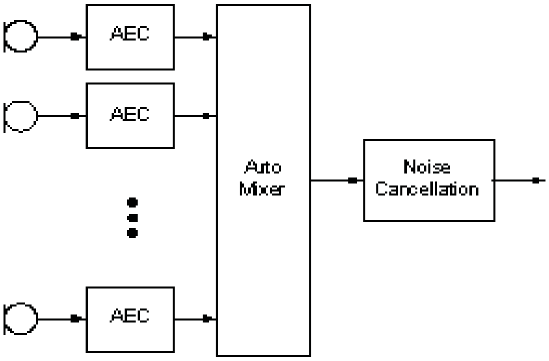
The non-linear suppression employed by acoustic echo cancellers is usually not included in the ERLE measurement as the suppression will make the ERLE appear much higher.

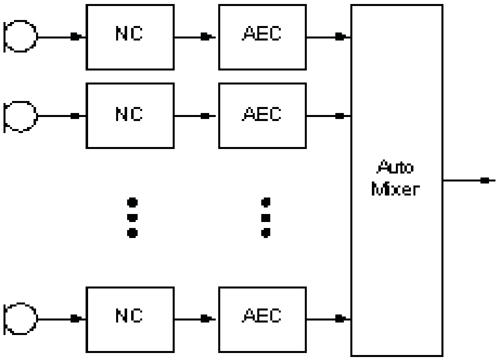


For some manufacturers of echo cancellers, the term ERL is replaced with an equivalent term of conferencing room gain. The conferencing room gain is the negative of the ERL, or defined as 10 Log (B/A). Lower room gain (equivalently higher ERL) in this case will improve the performance of the echo canceller.

Multi Channel vs. Single Channel AEC

The highest quality audio conferencing solution occurs when each microphone has its own acoustic echo canceller and noise canceller as shown on the right side of the following figure. It is clear that each microphone will pick up different background noise levels and possibly types of noise based on their proximity to different noise sources such as projectors, HVAC vents, lighting fixtures, etc. If each microphone is processed independently, then only the noise that is affecting a particular microphone will be removed from that microphone signal. This minimizes the likelihood of residual artifacts from the noise processing on the underlying signal. If the microphones are first summed together and then the noise processing is performed (as shown in the left side of the following figure), each local talkers' voice will be processed by the same noise reduction algorithm to remove noise regardless of whether that noise was incident on that particular microphone.





Similarly with respect to the acoustic echo canceller, it is also better to process the signals independently as each microphone has its own acoustic view of the room and sees an acoustic echo that is different from the other microphones. If multiple microphones are mixed into a single acoustic echo canceller, then as each microphone becomes active with respect to its automatic microphone mixing algorithm, the acoustic echo canceller must reconverge to the echo path from the microphone that becomes active. The performance of the acoustic echo canceller will degrade as the number of open microphones that feed into it is increased. Multiple microphones should only be fed into a single acoustic echo canceller if the echo return loss is high such as in acoustically well treated room or when microphones are physically separated from loudspeakers or their gating (becoming active or inactive) is closely controlled.

Similarly with respect to the acoustic echo canceller, it is also better to process the signals independently as each microphone has its own acoustic view of the room and sees an acoustic echo that is different from the other microphones. If multiple microphones are mixed into a single acoustic echo canceller, then as each microphone becomes active with respect to its automatic microphone mixing algorithm, the acoustic echo canceller must reconverge to the echo path from the microphone that becomes active. The performance of the acoustic echo canceller will degrade as the number of open microphones that feed into it is increased. Multiple microphones should only be fed into a single acoustic echo canceller if the echo return loss is high such as in acoustically well treated room or when microphones are physically separated from loudspeakers or their gating (becoming active or inactive) is closely controlled.

Muting Microphones

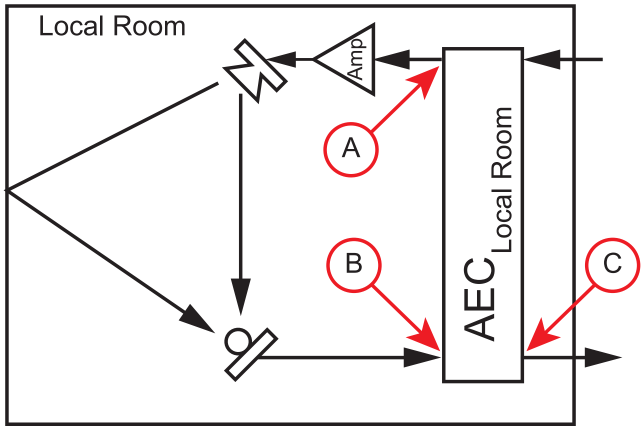
When muting microphones in an audio conferencing system, it is best to mute the microphones in the signal chain after the AEC has processed the local microphone's audio signal. This allows the acoustic echo canceller to continue to adapt to changes in the room, keeping the acoustic echo canceller converged even though the microphone is "muted".

If the microphones are muted in the signal chain before the acoustic echo canceller, then while the microphones are muted, the AEC will not be able to adapt as there will be no signal present. Once the microphones are unmuted, the AEC may have to reconverge to any new echo paths. This may result in a momentary acoustic echo to the remote site until the AEC reconverges.

Volume Control

As shown in following figure, if the gain on the loudspeaker amplifier is increased, the echo return loss will decrease meaning that there will be a more stronger echo at the input of the microphones. This means that the echo canceller will need to work harder to remove the echo. If the amplifier is turned up too much (sometimes by as little as 6 or 10 dB), the acoustic echo canceller will not be able to operate properly. This may result in persistent residual echo under low echo return loss (high echo) situations.

To properly adjust the volume of local room, one should adjust the sources feeding into the echo canceller (the far in signals) or if that is not practical, at least ensure that the echo canceller reference is also adjusted as the loudspeaker level is adjusted to match the adjustments of the amplifier. Under these conditions the ERL will remain relatively fixed, allowing the echo canceller to continue operating properly.



AEC Troubleshooting Guidelines

In the event that a configured conferencing system has acoustic echoes that are heard by the remote participants, the most important step in troubleshooting is to mute signal paths and determine when and if the echo goes away. If the echo goes away when the local microphones are muted, then the local echo canceller is causing the echo issue.

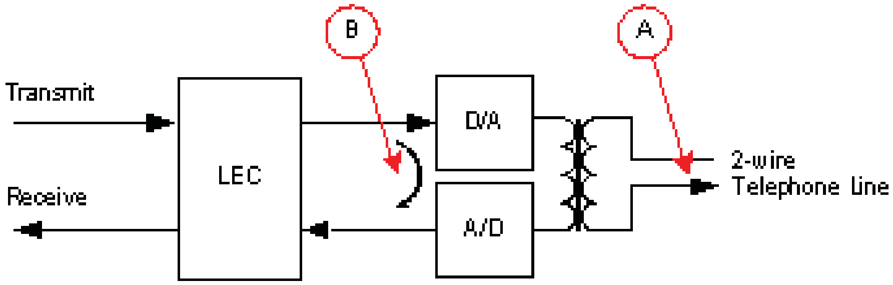
The most common reason for acoustic echo is that the echo return loss of the room is not high enough to allow the acoustic echo canceller to properly adapt to the remote audio. This is usually solved by reviewing the gain structure and turning down the amplifier and bringing up the signals that make up the echo canceller reference.

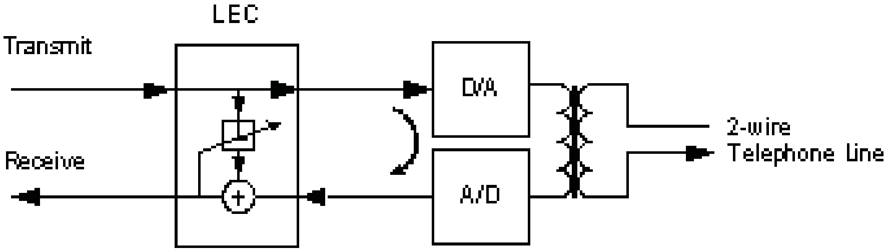
The next most common source of echoes is that the echo canceller reference does not contain all the remote audio sources, allowing one or more remote audio sources to be interpreted as local speech by the echo canceller and consequently sent to the remote participants.

Telephone Hybrid

To use the audio conferencing system, there must be a way to get the local signal to the remote participants and vice-versa. While only supporting 3.5 kHz of audio bandwidth, the Public Switched Telephone Network (PSTN) provides the most common and reliable real-time communication network for audio conferencing. In the conference room, the PSTN network is accessed by a pair of conductors that carry both the transmit and receive signals over the PSTN. When interfacing external equipment to the public switch network, it is necessary to separate the transmit and receive signals - this is the task of the telephone hybrid, also known as a 2-wire (PSTN) to 4-wire converter (separate transmit and receive signals).

The telephone hybrid circuit that interfaces the 2-wire PSTN network to the 4-wire separate transmit and receive uses a line echo canceller (LEC) which is similar to the acoustic echo canceller to remove line echo that is caused due to imperfect signal balancing of the transmit and receive circuits onto the 2-wire network. This imperfect balance means that when a transmit signal is sent to the telephone line, there is some leakage, or coupling, of the signal back to the receive path. This leakage is heard as a return echo of the local talker's speech. This is the same echo (commonly referred to as side-tone) that is heard on a telephone handset when speaking into a telephone - this side-tone echo serves the purpose of providing feedback to the local talkers as to how loud they are talking and that the phone line is working properly.





While side-tone is desirable while talking on a handset, it is not desirable in a conferencing application. As the line echo (or side-tone) is mixed together with the audio from the remote telephone talkers' speech, the line echo will be played into the room over the same loudspeakers in the local room used to hear the remote talkers. The line echo will sound like an echo of the local talkers' speech back to them with short delay. If the signal is loud enough in the room and there is significant loudspeaker to microphone coupling, such as low ERL, the line echo may cause acoustic feedback to occur in the room as the local talkers audio is played back into the room, picked up by the microphones reflected off the telephone line interface and played back into the room. Since the telephone signal is part of the acoustic echo canceller reference signal, the AEC will try to echo cancel the side-tone and prevent it from being sent to the remote side (and hence causing more side-tone) but it may not completely cancel the signal as the system will be in double-talk, meaning that the echo canceller will detect both the local talkers speech and the side-tone (interpreted as the remote talkers' speech). This will prevent the acoustic echo canceller from converging properly, degrading the conferencing experience.

The line echo canceller shown in the previous figure above is an adaptive filter that uses the transmit signal to adapt and remove the line echo that appears on the receive side of the line echo canceller. Similar to the acoustic echo canceller, there may also be some non-linear suppression to remove residual echoes when the line echo canceller is not fully converged. The line echo canceller typically is adapting when there is transmit audio present and no receive audio, in other words, just the local talkers are speaking. This is in contrast to the acoustic echo canceller which typically is only adapting when there is remote audio present and the local talkers are not talking.

Due to the variation in performance of telephone lines across the world, telephone hybrids will have a country code setting that will be required to properly interface the hybrid to the local phone line characteristics. If the phone hybrid settings don't match the communications network, the performance of the phone hybrid will be degraded, and possibly in violation of local telecommunications regulatory requirements.

Amplifiers

There are two broad classes of amplifiers - low impedance and constant voltage. The low impedance amplifiers are the type of amplifier used in consumer applications and the constant voltage amplifiers are used in larger, professionally installed systems.

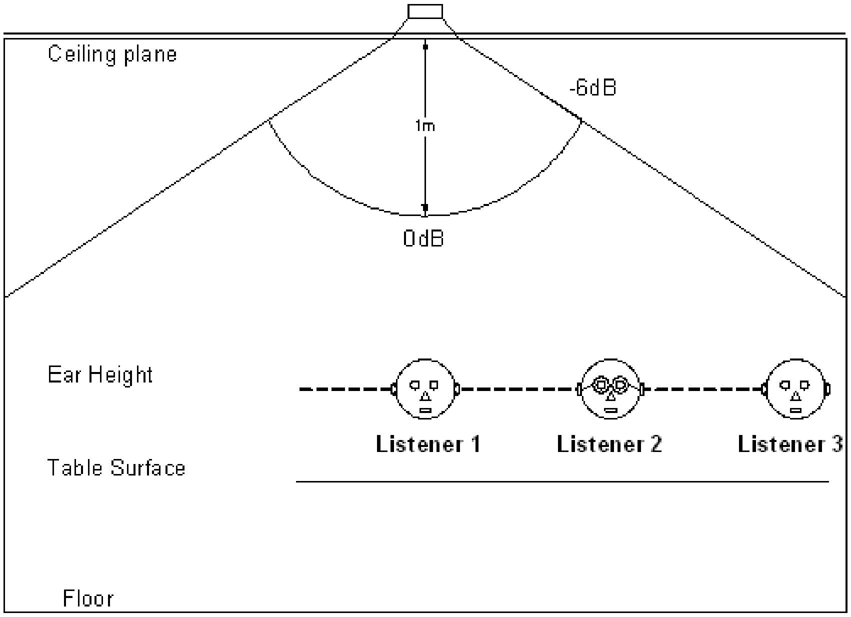
Low impedance amplifiers are designed to drive audio into low impedance loudspeakers typically with impedances between 4 and 16 ohms. These amplifiers are often used for smaller systems with one, two, three, or four loudspeakers and are suitable for use in medium to small conference rooms. In larger systems connecting all the loudspeakers will significantly reduce the impedance that the amplifier will see - potentially causing the amplifier to generate more current than it has been designed to provide. As more and more loudspeakers are connected to a low impedance amplifier, the impedance that the amplifier sees gets smaller and smaller which requires more and more current from the amplifier until the amplifier can not produce any more current and shuts down. In addition, if loudspeakers are removed or added to the system, they will affect the impedance of the remaining collection of loudspeakers, perhaps requiring changes to the volume levels to ensure that the playback signal is loud enough and the amplifier is still operating within its designed current range.

Constant voltage amplifiers, such as 70 V systems, make it easy to have large numbers of loudspeakers connected to the system as each loudspeaker can tap some power from the amplifier (using a transformer that is built into the loudspeaker) without being concerned as to the overall impedance that the amplifier sees. Large distributed loudspeaker systems are nearly always driven by constant voltage amplifiers.

Loudspeakers

Loudspeakers and the amplifier driving the loudspeakers in the local room allow the local conferencing participants to hear the remote audio. The loudspeakers can be wall-mounted, ceiling-mounted, or even table-mounted, although they are most often installed in the ceiling of the local rooms. As mentioned in the Amplifiers section, loudspeakers are either low impedance or 'transformer tapped' depending on the style of amplifier they will be connected to.

Just as microphones are characterized by their pickup pattern, frequency response, and sensitivity to sound, loudspeakers are characterized by their frequency response (80 Hz to 20 kHz typical), power capacity (40 to 80 Watts typical), sensitivity (86 dB SPL @ 1 m typical), and nominal coverage angle (130 degree typical). The loudspeaker coverage angle is defined by the angle where the loudspeaker levels are no less than 6 dB below the on-axis level. The following figure shows a typical coverage angle for a ceiling mounted loudspeaker and illustrates that listeners farther away from the loudspeaker axis will receive less sound than listeners directly below the loudspeaker. In the following figure, not only are listener 2 and listener 3 farther away from the loudspeaker than listener 1 and receive less audio due to the inverse square relationship, but they also receive less audio from the loudspeaker due to the inherent 6 dB difference between the off-axis response from the on-axis response of the loudspeaker.



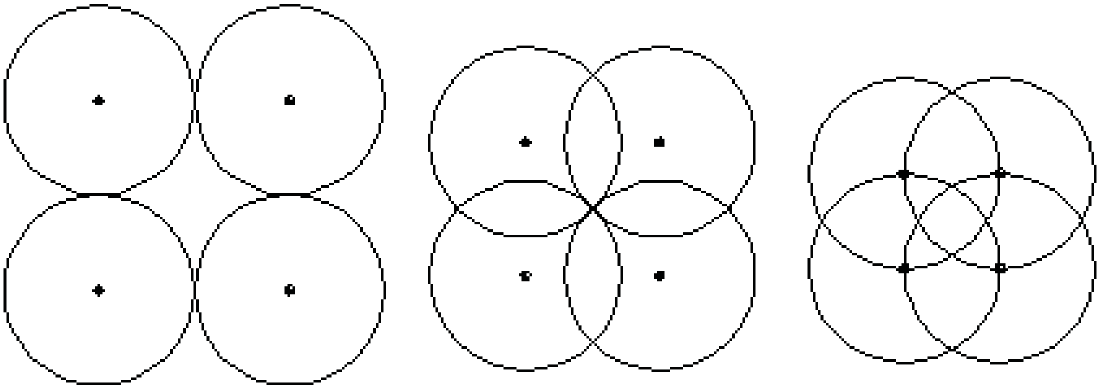
While this example shows a single loudspeaker and multiple participants, most rooms will require multiple loudspeakers to provide enough coverage that the remote audio can be heard at a comfortable level by all the participants, regardless of where they are sitting.

There are numerous rules of thumb for how many loudspeakers and the amount of power required in a particular size room and there are also software tools that are freely available to help with the power requirements and loudspeaker placement for a given sized room.

One rule of thumb is that the typical ceiling-mounted loudspeaker will have a coverage area of approximately 100 sq. feet at ear level assuming a 9 foot ceiling - shown in the following figure as a radius of about 5 ½ feet. Depending on how close the loudspeakers are clustered, there will be different levels of audio that are heard by local participants. As shown in the following figure, clustering the loudspeakers farther apart (about 11 feet in this example) will cover a larger area but have some significant level variation depending on where the listeners are. Clustering the loudspeakers closer together will increase the overlap and reduce the level variation that is heard by listeners based on their position in the room. While tighter clusters (such as the right hand of the following figure) will minimize the level variation, it will also increase the cost of the system as more loudspeakers will be required to cover a particular area. This is little benefit of increasing the number of loudspeakers beyond the tightest pattern shown in following figure where the loudspeakers are approximately 5½ feet apart.

Another rule of thumb about loudspeaker positioning with listeners is to distribute the loudspeakers no greater than twice the distance from the ceiling to the listener's ear level. In a conference room with 9 foot high ceilings and seated listeners' ears about 3 ½ feet above the floor, this rule of thumb corresponds to a

5½ foot from ceiling to ear distance which corresponds to an 11 foot loudspeaker separation from ceiling to ear. If the listeners will be standing, more loudspeakers will be required to effectively cover the room as the pattern of the loudspeaker will not cover as large an area closer to the loudspeaker.

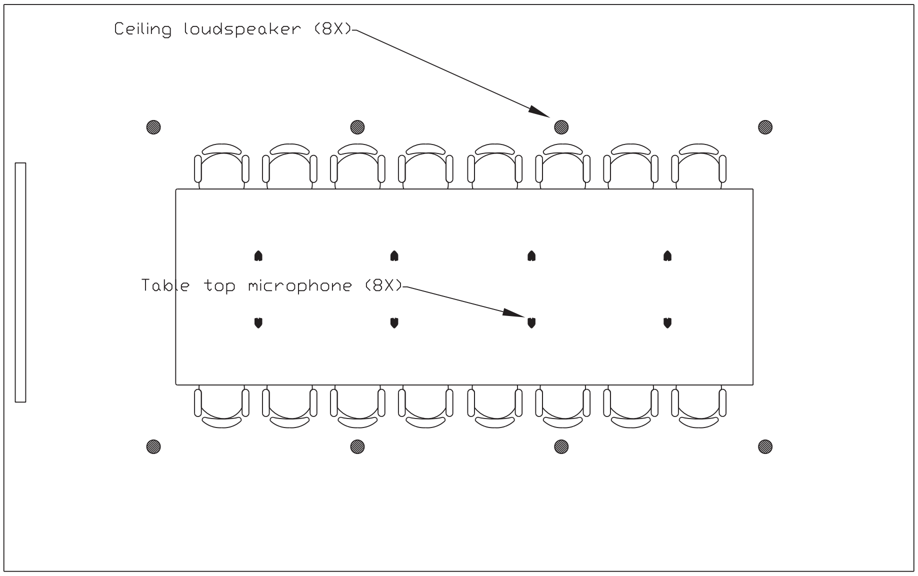


Distributing the loudspeakers in the ceiling allows for all participants in the room to hear the sound well. If there is only a single set of loudspeakers in the front of the room, then the audio in the back of the room may sound reverberant and muddy, reducing intelligibility for the local participants. Similar to a microphone's critical distance, loudspeakers also have a critical distance where the reverberant sound field is equal in intensity to the direct sound field. If there are few loudspeakers (such as positioned at the front of a room), then listeners near the critical distance will not receive an intelligible signal. By distributing the loudspeakers throughout the room, it is generally possible to ensure that all listeners are well within the critical distance of the loudspeakers.

In many audio conferencing rooms, there are separate loudspeakers for program audio such as VCR or DVD to maintain the stereo separation of the source material in the room. There can also be a subwoofer for this media, although subwoofers are typically not required for audio conferencing due to the limited range of low frequencies produced by the typical human talker.

Speaker Zoning And Placement

As discussed in the previous section, ceiling loudspeakers should be positioned so the sound field from the loudspeaker encompasses the participants' seating locations. This typically means that the loudspeakers are positioned outside of the table perimeter to allow coverage of the participants and any potential areas of overflow seating as shown in one possible layout of a room that is approximately 33x20 feet (approximately 700 sq. ft.) in the following figure. As discussed in an earlier section, the microphones should be placed as close to the local participants as possible to minimize the amount of background noise and reverberation.


Ceiling loudspeaker (8X)
Table top microphone (8X)

The next figure shows the room of the previous figure with the directional microphone pickup patterns and a 100 sq. ft. loudspeaker pattern overlaid. Notice that the microphones are at the periphery of the loudspeaker coverage pattern (to maximize the ERL of the room) and that while there is not 100% loudspeaker coverage over the center of the table there is adequate coverage of the audio in the room for both the seated participants and any overflow seating for extra participants.

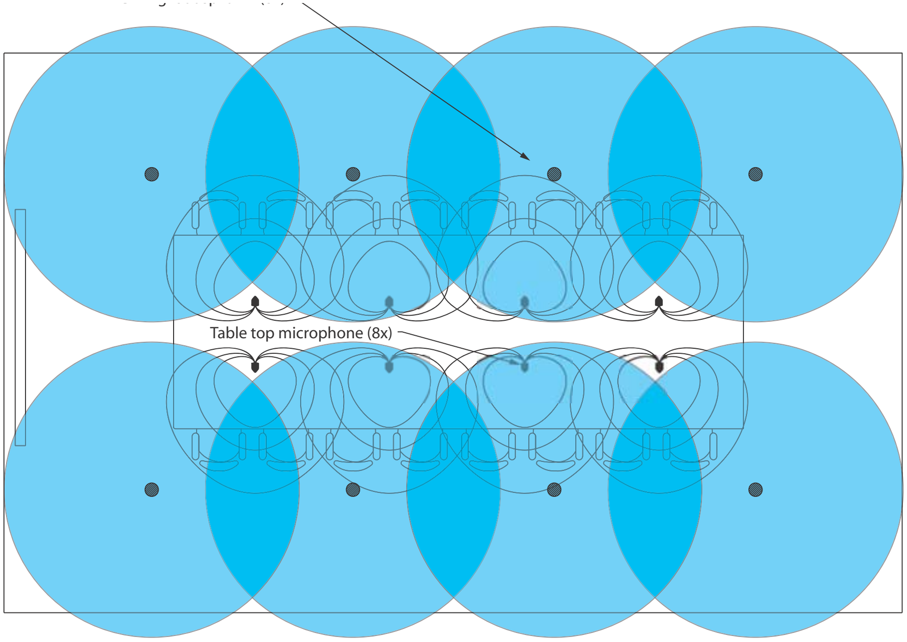


Table top microphone (8x)

Loudspeakers - How Much Power Is Required

Once the locations of the loudspeakers have been determined, the next step is to determine how much power is required to drive each loudspeaker to achieve the required level at the listeners. Loudspeakers have a power rating that is expressed in dB SPL @ 1 meter. This specifies how much sound is created at 1 m from the loudspeaker assuming 1 W of power is applied to the loudspeaker. For instance a loudspeaker rated at 86 dB SPL 1 Watt/meter requires 1 W to generate 86 dB SPL at a distance of 1 m from the loudspeaker. In a normal room, the loudspeakers are approximately 2 m above the participant that is seated directly underneath a loudspeaker, resulting in 80 dB SPL at the listener's location with 1 W of power. As normal speech conversations are in the 70 dB to 77 dB SPL range, this level of sound would be more than sufficient for playback for a listener located directly underneath a loudspeaker assuming there is not a significant amount of background noise in the room (such as in an NC 30 style room). As mentioned earlier, having a sound system that is at least 25 dB above the ambient background will yield excellent intelligibility.

To compute the power requirements for each of the distributed loudspeakers, it is convenient to take the desired SPL at the listeners' ears (for example 85 dB SPL), add additional level to compensate for the distance from the loudspeakers (for example listeners 2 m away from the loudspeaker will require an additional 6 dB to compensate for the 6 dB loss 2 m away from the loudspeaker), and subtract the loudspeakers sensitivity (86 dB SPL). In this example this results in 85 dB + 6 dB - 86 dB = 5 dBW. This is the power in dBW (referenced to 1 W). To convert this back to Watts, it is necessary to take the inverse log (recall that a power in dBW is 10 log (Power in watts/1 watt)). The result is that 3.16 W is required for each loudspeaker. As most constant voltage systems have a switch to select the appropriate power setting, selecting the tap closest to 3 W would generate the proper level for the loudspeakers.

Once the power requirements have been calculated for the loudspeakers, add them up and use this as the baseline for the size of the amplifier required to drive audio into the room. Keep in mind that there will be transformer losses on the order of a 1 dB or so for each loudspeaker, so choose an amplifier at 25% to 50% larger than the sum of the required loudspeaker power. This will allow some headroom for additional loudspeakers or louder transient signals.

Spatial Directionality

In video conferencing applications, it is common to have the remote audio come from a location close to the display screen to give the perception that the remote audio is coming from the remote participants shown on the video screen. In this situation, when the room is larger than about 20 ft. in length, reinforcement of the front speakers with additional ceiling loudspeakers distributed through the room will better fill the room with sound (keeping all listeners well within the critical distance of the loudspeakers) without losing the perceptual directionality associated with the front of room display device. To maintain the perception of the audio coming from the front of room display device, the reinforcement of the ceiling loudspeakers can be delayed slightly (1 msec for each foot of separation) from the front loudspeakers and can be attenuated by approximately 6 - 10 dB from the level sent to the front of room loudspeakers.

Microphone And Loudspeaker Placement Considerations

Once loudspeakers have been placed through the room to achieve good sound coverage, and microphones have been placed through the room to provide good sound pickup, it is still necessary to double check the placement of the microphones relative to the loudspeakers to ensure the ERL of the conferencing system hasn't been compromised in the process. If loudspeakers are placed in the direct pick up pattern of the microphone, the ERL will be reduced and there is a higher chance of residual echo being sent to the remote participants. Typically the configuration software with the audio conferencing product will provide a tool where the ERL can be checked to ensure it is within the recommended operating range for that particular audio conferencing device (typically 0 to 10 dB). If the recommended ERL is exceeded, it will be necessary to review the gain structure (lowering the loudspeaker amplifier settings and increasing the remote audio levels), ensuring the loudspeakers are pointed directly into the pick up pattern of the microphones, or lowering the gain on the microphones.

In-Room Reinforcement

In a given room, sound reinforcement may be desirable if it is not possible to have listeners easily hear other talkers in the room. The objective of this type of sound reinforcement (sometimes referred to as "voice lift" in the generic sense) is to augment the local talker's voice so that local listeners still have the perception that the audio is originating from the local talker but now the level of the speech has been increased subtly via the installed loudspeaker system.

One of the keys to making a system of this type stable and work reliably is properly setting user expectations and perception. It is critical that the users do not expect audio levels similar to a “paging system” or “public address” (PA) system. Paging and PA systems are designed to broadcast a single voice loudly, overcoming background noise, side conversations, and inattention in order to deliver important messages. The sound levels generated by these systems are much louder than local participants in a room would ever need to subtly reinforce their voice and maintain the perceptual directionality so the local listeners' attention remains focused on the talker and not on the loudspeaker.

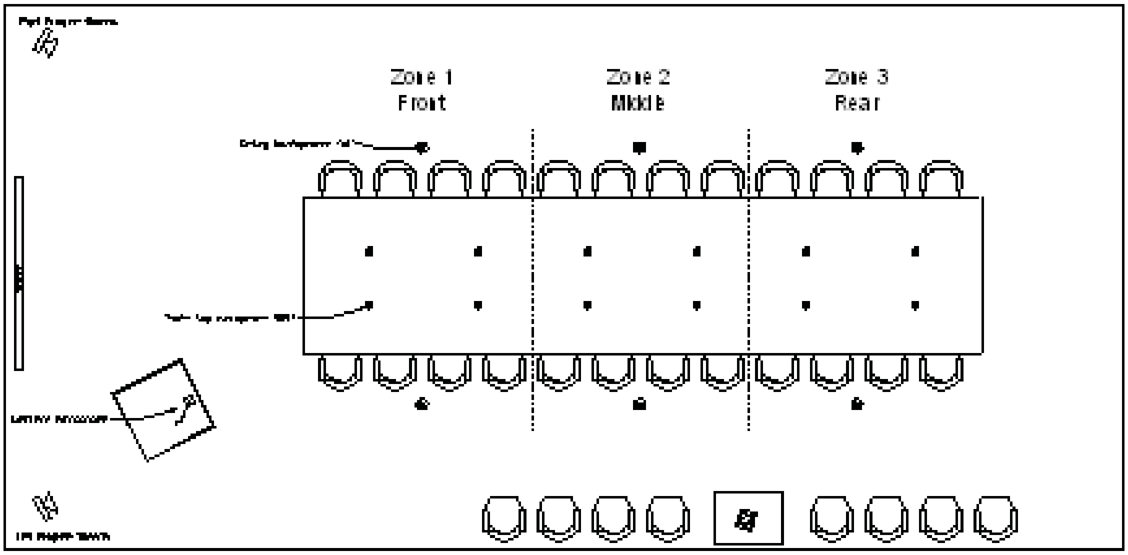
Voice reinforcement or enhancement is supposed to be just that; reinforcing the level of the talker to improve intelligibility and not re-create the sound level of an emergency page. In a properly configured and calibrated voice lift system, the best way to find out if the system is functioning properly is to disable it and see if the local participants notice the resulting loss of intelligibility. If they do, then the system is set correctly. The reinforced levels should never exceed conversational speech levels (approximately 70 dBA SPL typical at the listener's ear) or the result may become unstable, creating residual echoes to the remote listeners due to low ERL and worse may generate very loud acoustic feedback in the room with loudspeaker audio coupling into the local microphones.

Voice lift is typically needed when the room is large enough that the local talkers' audio drops below 60 dBA SPL when heard by the farthest listener in the room. Based on normal speech conversation levels of 70 - 77 dBA SPL this roughly corresponds to listeners that are approximately 20 to 25 feet away from the talker. Another way to view this is as a critical distance issue. If the local listeners are farther away than the critical distance form the local talker, some form of voice lift may be required for adequate intelligibility.

For rooms smaller than approximately 20 feet x 20 feet, the voice lift application is really not necessary and is just starting to be of some benefit in rooms 30-feet (9 meters) square. Just as adding gain to a microphone to try to compensate for a critical distance issue does not work, adding sound reinforcement to compensate for a noisy room doesn't work well either. The microphones that pick up the noise will reinforce that noise into the room, adding to the noise rather than making it easier to be heard. The correct approach would be to get rid of the noise and make the room useful in all applications.

So, how does one realistically go about making this system work? One must be careful planning microphone and speaker locations, so that the system stays acoustically stable (that is, no squealing and howling of feedback as different microphones come active) while providing the necessary pickup of local talkers and reducing the "effective acoustic distance" (how far away do they sound) of the listeners. First, consider the number of "zones" or areas of independent loudspeaker playback. A "zone" is most often sized close to the largest room size that does not need reinforcement, or about 20-feet square. A room 20-feet by 40-feet would be 2 zones, one 40-foot square would be 4 zones and so on. Long, narrow boardrooms will often require this type of voice reinforcement application.

Once the “zones” are planned, the microphone locations are selected within those zones. The concept here is to locate microphones and speakers such that each zone is completely independent in level and mix. This way, microphones from a given zone are never played into the loudspeakers associated with that same zone (mix-minus) and are sent at increased levels to zones further away (the inverse square law calculated results drives the required level settings in the reinforcement system - zones twice as far away will typically have 6 dB more level). To support zoning, a multi-channel amplifier must be used so that each loudspeaker zone can receive separate loudspeaker signals.



There are two general concepts that are often used in voice lift. Needed Acoustic Gain (NAG) or "how loud does it need to be" and Potential Acoustic Gain (PAG) or "how loud can it be without feedback" are the calculations that can be done with a few different online tools, and will quickly help determine the sound levels that can be tolerated within a room. For the room to be acoustically stable, the NAG must be less than the PAG, and in fact should be less than PAG by some safety margin just to be safe.

Occasionally the PAG can be slightly improved with equalization, feedback eliminators (mostly these are just fast reacting narrow filters that reduce the gain at the ringing signal at the onset of feedback), and microphone/loudspeaker directionality improvements, but those are usually limited to less than 6 dB total improvement. Placing microphones as close as possible to the local talkers, and minimizing the number of active microphones will help with NAG/PAG.

Ceiling mounted microphones present particularly difficult sound reinforcement challenges due to their close proximity to loudspeakers (decreasing PAG) and their long distance from the local talkers (increasing NAG). The use of ceiling microphones and sound reinforcement must be designed extremely carefully with conservative levels of reinforcement, large separation of zones, and limited volume control range to ensure that the resulting room is stable. In situations where ceiling microphones and loudspeakers must be used with the added requirement of in-room reinforcement, it is recommended that the design and installation be performed by a professional who specializes in these applications.

In a room that has sound-reinforcement with inappropriately high gain settings, there is no longer any such thing as a "side conversation". Everyone in the room will likely be able to hear all conversations, making it impossible to have side comments that are private.

With reinforcement applications, remember that the key to success is setting the appropriate performance expectations with the end user and it is the responsibility of the conference room designer to set that expectation.